ISSN: 2321-7782 (Online)
Volume 2, Issue 10, October 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

A Literature Survey on Security Evaluation of Pattern Classifiers under Attack

Dr. Amitabh Wahi[1]
Professor/ Dept. of IT,
Bannari Amman Institute of Technology,
Sathyamangalam

C.Prabakaran[2]
PG-scholar/Dept.of IT,
Bannari Amman Institute of Technology,
Sathyamangalam

Abstract: ***Pattern classification systems are commonly used in adversarial applications, like biometric authentication, network intrusion detection, and spam filtering, in which data can be purposely manipulated by humans to undermine their operation. As this adversarial scenario is not taken into account by classical design methods, pattern classification systems may exhibit vulnerabilities, whose exploitation may severely affect their performance, and consequently limit their practical utility. Extending pattern classification theory and design methods to adversarial settings is thus a novel and very relevant research direction, which has not yet been pursuedin a systematic way. The system evaluates at design phase the security of pattern classifiers, namely, the performance degradation under potential attacks they may incur during operation. A framework is used for evaluation of classifier security that formalizes and generalizes the training and testing datasets.***

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

I. INTRODUCTION

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 classes (e.g., legitimate and spam emails). 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 (spoofing attack) [1], [2]; modifying network packets belonging to intrusive traffic to evade intrusion detection systems (IDSs) [3]; manipulating the content of spam emails to get them past spam filters (e.g., by misspelling common spam words to avoid their detection) [4], [5], [6].

II. FEASIBILITY OF ATTACKS

Spoofing Attack

Biometric systems have been found to be useful tools for person identification and verification. A biometric characteristic is any physiological of behavioral trait of a person that can be used to distinguish that person from other people. A few key aspects of a human physiological or behavioral trait that make for a strong biometric for recognition are universality, distinctiveness, permanence, and Collectability. These ensure that the trait is available from all people, is adequately variable among all people, does not change significantly over time, and is reasonably able to be measured. The problem with any human trait that meets these criteria is in the performance, acceptability, and circumvention of the biometric feature. Performance is an issue resulting mainly from the combination of lack of variability in the biometric trait, noise in the sensor data due to environmental factors, and robustness of the matching algorithm. Acceptability indicates how willing the client pool will be to use the biometric identifier regularly. Circumvention is the possibility of a non-client (impostor) getting past the system using deceptive methods.

Typically these methods involve the forgery of the biometric trait, an act commonly termed "Spoofing". A multi-biometric system is one that combines information from multiple sources in an attempt to reduce the effect of poor performance in any one source. Multi-biometric systems have commonly taken three forms; single biometric trait multiple representation, single biometric trait multiple matcher, and multiple biometric trait. These three methods seek to reduce errors due to noisy sensor data, poor matcher performance, and poor performance in a biometric trait in general. Implementing multiple modalities (i.e., biometric traits) in a system, for instance, face, iris, and fingerprint, requires an imposter to spoof more than one biometric trait, making it much more difficult to fool the system. This gives multimodal systems a leading edge over the other two classes of multi-biometric systems in terms of security.

The key to creating a secure multimodal biometric system is in how the information from the different modalities is fused to make a final decision. There are two different categories of fusion schemes for multiple classifiers; rule based and supervised based. Supervised methods, on the other hand, require training but can often provide better results than the rule based methods. For example, have shown that a fusion strategy using a support vector machine (SVM) was able to out-perform a fusion algorithm using the sum rule. Introducing a quality measure into a fusion algorithm is one method that has been used to boost performance in multi biometric systems.

If for instance, a more secure biometric of high quality gives a low match score and a less secure biometric gives a high match score, then there is a high likelihood of a spoof attack. It is commonly understood that one of the strengths of a multimodal system is in its ability to accommodate for noisy sensor data in an individual modality. In contrast, a more secure algorithm, in order to address the issue of a spoof attack on a partial subset of the biometric modalities, must require adequate performance in all modalities. This type of algorithm would invariably negate, to some extent, the contribution of a multimodal system to performance in the presence of noisy sensor data. A multimodal system improves the performance aspect but increases the security only slightly since it is still vulnerable to partial spoof attacks. Enhanced fusion methods, which utilize approaches to improve security, will again suffer decreased performance when presented with noisy Data.

III. SPAM AND ONLINE SVMS

The support vector machine (SVM)is a exercise procedure for knowledge organization and reversion rubrics after statistics, for instance the SVM can be recycled to study polynomial, circular foundation purpose (RBF) then multi-layer perception (MLP) classifiers SVMs remained chief optional by Vapnik in the 1960s for organization beside smustlately develop an part of penetrate in investigate on owed to growths in the methods plus philosophy joined with postponements to reversion and thicknessapproximation.SVMsascendedafterarithmeticalknowledgephilosophy the goal existence to resolve separate the problematic of attention deprived of resolving additional problematic as an middle stage. SVMs are founded on the physical threat minimisation code, carefully connected to regular inaction philosophy. This belief joins volume switch to stop over-fitting and therefore is ain complete response to the bias-variance trade-off quandary. Binary key rudiments in the application of SVM are the methods of precise software design and seed purposes. The limits are originated by resolving a quadratic software design problematic with direct parity and disparity restraints; slightly than by resolving a non-convex, unimpeded optimisation problem. The suppleness of seed purposes lets the SVM to exploration a extensive diversity of theory places. The geometrical clarification of support vector classification (SVC) is that the procedure pursuits for the best unravelling superficial, i.e. the hyper plane that is, in a intelligence, intermediate after the binary courses.

This best unscrambling per plane has several agree able arithmetical possessions SVC is drawn chief aimed at the linearly divisible circumstance. Kernel purposes are then presented in instruction to concept non-linear choice exteriors. In conclusion, for noisy data, when whole parting of the binary courses might not be desirable, relaxed variables are presented to permit for exercise faults.

IV. SPAM FILTERING OVERVIEW

Over the past few years, spam filtering software has gained popularity due to its relative accuracy and ease of deployment. With its roots in text classification research, spam filtering software seeks to answer the question "Whether the message x is spam or not?". The means by which this question is addressed varies upon the type of classification algorithm in place. While the categorization method differs between statistical filters, their basic functionality is similar. The basic model is often known as the bag of words (multinomial) or multivariate model. Essentially, a document is distilled into a set of features such as words, phrases, meta-data, etc. This set of features can then be represented as a vector whose components are Boolean (multivariate) or real values (multinomial). One should note that with this model the ordering of features is ignored. Classification algorithm uses the feature vector as a basis upon which the document is judged. The usage of the feature vector varies between classi_cation methods. As the name implies, rule based methods classify documents based on whether or not they meet a particular set of criteria. Machine learning algorithms are primarily driven by the statistics (e.g. word frequency) that can be derived from the feature vectors. One of the widely used methods, Bayesian classification, attempts to calculate the probability that a message is spam based upon previous feature frequencies in spam and legitimate e-mail.

V. SPAM FILTERING TECHNIQUES

A naive Bayes filter describes the joint probability over a set of features X = X1, X2, . . . ,Xn and a class C by making independent and the class given as:

$$\mathbf{P(X, C) = P(C)\ \pi P(Xi|C)}$$

The class prior, P(C), represents the relative frequency of each class, while the conditional probabilities P (Xi|C) encode the probability of each feature value given the class. In the spam domain, the class prior P(spam)(where P(legit) = 1 − P(spam)) represents what fraction of incoming email is spam, and the probabilities P(word|spam) and P(word|legit) give the probability that each type of email contains each word (or other feature).Classification follows from the class posterior odds:

$$\frac{\mathbf{P(spam|X)}}{\mathbf{P(legit|X)}} = \frac{\mathbf{P(spam)}\prod_{\mathbf{i=1}}^{\mathbf{n}}(\mathbf{P(Xi|spam)})}{\mathbf{P(legit)}\prod_{\mathbf{i=1}}^{\mathbf{n}}(\mathbf{P(Xi|legit)})}$$

If chances out do some dawn, then the email is confidential as junk. By taking the logarithm, log odds can be represented as the following sum:

$$\log\left(\frac{\boldsymbol{P(spam)}}{\boldsymbol{P(legit)}}\right) + \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{log}\left(\frac{\boldsymbol{P(Xi|spam)}}{\boldsymbol{P(Xi|legit)}}\right)$$

Features that are more common in legitimate email than spam have negative weights, while those more common in spam have positive weights. Unlike naive Bayes, which models the joint distribution of all features, P(X, C), a maxent filter models the conditional probability of the class given the other features, P(C|X). Assuming Boolean features and two classes (spam and legit), the maxent filter is a logistic regression model and the class odds take the following exponential form:

$$\frac{\boldsymbol{P(spam|X)}}{\boldsymbol{P(legit|X)}} = \mathbf{exp}(\sum_{\boldsymbol{i=0}}^{\boldsymbol{n}} \boldsymbol{\lambda i Xi})$$

Where X0 is an artificial feature that has the value one for every message. As with naive Bayes, the classification isdetermined by comparing this sum with a threshold. Naive Bayes and maxent share the same representation (weights and a threshold), but are trained to optimize different criteria. In this way, maxent filters are similar to support vector machines: both model the class boundary rather than the entire instance space.

VI. NAÏVE BAYES SPAM FILTERING

Naive Bayes classifiers are a popular arithmetical technique of e-mail filtering. They typically use bag of words features to identify spam e-mail, an approach generally used in text classification. Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other things), with spam and non-spam e-mails and then using Bayesian suggestion to calculate a probability that an email is or is not junk. Naive Bayes spam filtering is a baseline method for dealing with spam that can tailor itself to the electronic mail needs of individual users and give low false positive spam detection rates that are normally acceptable to users.

A. Process

Particular words have particular likelihoods of occurring in spam email and in genuine email. The filter doesn't know the probabilities in advance. So it must first be trained and so it can build them up. To train the filter, the user must actually sign post whether a new email is spam or not. As in any other spam filtering practice, email marked as spam can then be mechanically moved to a "Junk" email folder, or even deleted outright. Some software implements quarantine devices that define a time surround during which the operator is allowed to review the software's decision. The initial training can usually be sophisticated when wrong judgment's from the software are known (false positives or false negatives). That allows the software to energetically adapt to the ever developing nature of junk. Some junk filters syndicate the results of both Bayesian junkriddling and other heuristics and following in even advanced pricking accuracy, occasionally at the cost of adaptiveness.

The formula used by the software to control that is derived from Bayes' theorem where:

$$\mathbf{Pr}(\boldsymbol{S}|\boldsymbol{W}) = \left(\frac{\boldsymbol{Pr}(\boldsymbol{W}|\boldsymbol{S}).\boldsymbol{Pr}(\boldsymbol{S})}{\mathbf{Pr}(\boldsymbol{W}|\boldsymbol{S}).\mathbf{Pr}(\boldsymbol{S}) + \mathbf{Pr}(\boldsymbol{W}|\boldsymbol{H}).\mathbf{Pr}(\boldsymbol{H})}\right)$$

- Pr(S|W)is the probability that a message is a spam, knowing that the word "replica" is in it;
- Pr(S)is the overall probability that any given message is spam;
- Pr(W|S)is the probability that the word "replica" appears in spam messages;
- Pr(H)is the overall probability that any given message is not spam (is "ham");
- Pr(W|H)is the likelihood that the word "replica" seems in ham mails.

B. Spamicity of a word

Recent numbers show that the current probability of any message being spam is 80%, at the very least:

$$\mathbf{Pr(s)=0.8;Pr(H)=0.2}$$

Though, greatest bayesian spam finding software makes the declaration that there is no *a priori*aim for any inwardmessage to be spam rather than ham, and forestalls both cases to have equal likelihoods of 50%.

$$\mathbf{Pr(S)=0.5;Pr(H)=0.5}$$

The filters that use this suggestion are said to be "not biased", meaning that they have no preconception regarding the inbound email. This assumption certifications simplifying the general formula to:

$$\mathbf{Pr}(\mathbf{S}|\mathbf{W}) = \frac{\mathbf{Pr}(\mathbf{W}|\mathbf{S})}{\mathbf{Pr}(\mathbf{W}|\mathbf{S}) + \mathbf{Pr}(\mathbf{W}|\mathbf{H})}$$

The number Pr(W|S)used in this formula is advanced to the incidence of mails containing "replica" in the mails recognized as spam throughout the knowledge stage. Likewise, Pr(W|H)is loomed to the incidence of mails greedy "replica" in the letters branded as ham throughout the knowledge stage. For these guesses to make sense, the set of scholarly messages needs to be big

and demonstrative enough. It is also sensible that the cultured set of mailsimitates to the 50% theory about repartition amid junk and ham, i.e. that the datasets of junk and ham are of similar cope.

C. Dealing with rare words

In the case a word has never been met during the learning phase, both the numerator and the denominator are equal to zero, both in the general method and in the spamicity formula. The software can choose to abandon such arguments for which there is no indication obtainable. More usually, the words that remained encountered only a few periods throughout the knowledge stage reason a problematic, since it would be an mistake to trust sightlessly the info they deliver. A simple answer is to just evade captivating such faulty arguments hooked on explanation as healthy. Smearing over Bayes' proposition, and presumptuous the categorization amid junk and ham of the communications surrounding a assumed term ("replica") is a chance mutable with beta delivery, certain packages choose to use a modified likelihood:

$$\boldsymbol{Pr'(S|W) = \frac{s.\mathbf{Pr}(S) + n.\mathbf{Pr}(S|W)}{s+n}}$$

where:

- Pr'(S|W)is the corrected probability for the message to be spam, knowing that it contains a given word ;
- S is the *strength* we give to background information about incoming spam ;
- Pr(S)is the probability of any incoming message to be spam ;
- n is the number of occurrences of this word during the learning phase ;
- Pr(S|W)is the spamicity.

VII. 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 [6], [12], [13], and it is included in several real spam filters.7 In this example, we focus on model selection. We assume that the designer wants to choose between a support vector machine (SVM) with a linear kernel, and a logistic regression (LR) linear classifier. He also wants to choose a feature subset, among all the words occurring in training emails. A set D of legitimate and spam emails is available for this purpose. We assume that the designer wants to evaluate not only classifier accuracy in the absence of attacks, as in the classical design scenario, but also its security against the well-known bad word obfuscation (BWO) and good word insertion (GWI) attacks. They consist of modifying spam emails by inserting "good words" that are likely to appear in legitimate emails, and by obfuscating "bad words" that are typically present in spam [6]. The attack scenario can be modeled as follows.

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

Knowledge: As in [6], [10], the adversary is assumed to have perfect knowledge of the classifier, i.e.,: (k.ii) the feature set, (k.iii) the kind of decision function, and (k.iv) its parameters (the weight assigned to each feature, and the decision threshold). Assumptions on the knowledge of (k.i) the training data and (k.v) feedback from the classifier are not relevant in this case, as they do not provide any additional information.

Capability: We assume that the adversary: (c.i) is only able to influence testing data (exploratory attack); (c.ii) cannot modify the class priors; (c.iii) can manipulate each malicious sample, but no legitimate ones; (c.iv) can manipulate any feature value (i.e., she can insert or obfuscate any word), but up to a maximum number n max of features in each spam email [6], [10].

This allows us to evaluate how gracefully the classifier performance degrades as an increasing number of features is modified, by repeating the evaluation for increasing values of n max.

VIII. CONCLUSION

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 forth is 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. 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 [12], [10]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.

References

1. R.N. Rodrigues, L.L. Ling, and V. Govindaraju, "Robustness ofMultimodal 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 Vulnerabilityto Non-Zero Effort (Spoof) Imposters," Proc. IEEE Int'lWorkshop 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 SecuritySymp., 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 SpamFilters," Proc. Second Conf. Email and Anti-Spam, 2005.
6. A. Kolcz and C.H. Teo, "Feature Weighting for Improved ClassifierRobustness," Proc. Sixth Conf. Email and Anti-Spam, 2009.
7. D.B. Skillicorn, "Adversarial Knowledge Discovery," IEEE IntelligentSystems, vol. 24, no. 6, Nov./Dec. 2009.
8. D. Fetterly, "Adversarial Information Retrieval: The Manipulationof 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. C_ardenas 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. 641- 647, 2005.

