A Comprehensive Survey on Pattern Classifier's Security Evaluation under Attack

Suryakant R. Paralkar
M. Tech Student,
CSE department, SGGSIE&T,
Nanded-431606, India.

U. V Kulkarni, Ph.D
Professor,
CSE Department, SGGSIE&T,
Nanded-431606, India.

ABSTRACT
From many years, various tools based on pattern classification system have been used in security related applications like, spam filtering, biometric authentication system, network intrusion detection system, pattern classification systems are used, in which brilliant and adaptive adversary may changes data to make the classifier produce false negatives (regular). Measurement of pattern classifier security performance is very important part for making decisions, finding product viability, for differentiate various classifiers. Pattern classification systems may exhibit obligations, exploitation affect their performance, produce limitations to practical utility, if adversarial scenario is not taken into account. At design phase, the system evaluates the classifiers security. The classifiers security means performance degradation for related attacks may acquire when operation runs. A phenomenon is used for classifier security evaluation, adversary model for defining attack scenarios that generates training and testing sets.

General Terms
Pattern classification, adversarial classification, performance evaluation, security evaluation, robustness evaluation.

Keywords
Legitimate samples, malicious samples, reactive and proactive arms race, spoof attack.

1. INTRODUCTION
Basically, in machine learning algorithms, pattern classification systems are used in security related applications like, spam filtering, biometric authentication system, network intrusion detection system, have been typically faced as two class classification problems, in which a classifier aims to differentiate between "legitimate" and "malicious" samples. In standard pattern classification theory, these applications are distinct, because they are defined by presence of brilliant adversary that generates malicious samples. For e.g., the target of biometric authentication system is to differentiate between registered user and malicious users, to grant or ban access to some stored private resources. Similarly, intrusion detection systems (IDSs) targets at differentiate between legitimate and malicious network traffic, and attackers may hide their network samples, so that they are mislabeled as legitimate. These applications exhibit adversarial nature, so that the data is actively manipulated by an adversary seeking to make the classifier produce false negatives (regular). According to that, a distinct design procedure is required to explicitly deal with the arms race existing in security applications between system designers and adversaries.

Basically, attacks against pattern classifiers are: submitting a fake fingerprints to biometric authentication system to gain access to system as a registered user (spoof attack) [13], [21], modification of network packets belongs to intrusive traffic to evade intrusion detection system [14], modification of spam emails by adding some words which are likely to appearing legitimate emails but not in spam and by obfuscating typical common spam words [3], [9], [17]. The adversarial applications are found in adversarial knowledge discovery [18], adversarial information retrieval [19]; e.g., a fraud webmaster take the charge of search engine and drive the website according to his orders. The protocol of network packets can be useful to improve the quality of service over a network in network protocol verification automatically recognizing. The adversarial performance of pattern classifier produces three queries-

- Accepting the pattern classifier systems obligations
- Solving pattern classifier security for related attacks
- Making pattern classifier system sturdy to attack

Basically, pattern classifier systems can be used in automatic web page ranking to automatically score or label pages conforming to predetermined topics. The adversarial pattern classification problem can be described as game between brilliant adversary and classifier designer. The game can be played in such a way that, they can broadcast the information about the game that each player has.

Rest of the paper is described as –section 2 describes background and related task, section 3 describes the framework for pattern classifiers security evaluation. First, to observed attacks, traced security in the position of arms race is not good. Second, defined adversary model in terms of adversary goal, knowledge, capability, for giving efficient guidance to realistic attack scenarios. Third, defined data distribution model, because targeted attacks effect on training and testing distribution. Section 4 gives acknowledgment. Section 5 provides the conclusion about the system.

2. BACKGROUND
This section provides the information about background and previous things.

2.1 Taxonomy of Attacks
Taxonomy of attack was defined in [8], [16] and explained in [6]. Attack taxonomy of pattern classifiers is based on three characteristics: type of attacks influence on pattern classifiers, type of security violation that they form, and attacks specificity.

2.1.1 Types of Attack Influence
2.1.1.1 Causative
In this attack influence, attackers target is to introduce vulnerabilities by making changes to the training data. If attacker can make spam to slip past the classifier as false negatives, then attacker can take the charge over training data. This type of attack influence makes effect on testing as well as training data or only on training data. To interfere with

operations like mailing by blocking mail, attacker uses control over training data.

2.1.1.2 Exploratory

In this attack influence, attackers target is to trace out vulnerabilities at the stage of classification. These attacks use other techniques like, detector probing, information discovery related to it or its training data, since they do not adjust the training samples. This type of attack influence makes effect only on testing data. Without direct influence over classifier itself, attacker crafts spam so as to evade classifier. In this attack, attacker has no control over learning or training data, but wishes to cause denial of service.

2.1.2 Types of Security Violation

2.1.2.1 Integrity

In integrity violation, attackers target is to get malicious samples being classified as legitimate. If the services or resources protected by the classifier are allowed by the adversary to access, then it is an integrity violation.

2.1.2.2 Availability

In availability violation, attackers target is to increment the classification improper value of legitimate samples. Classification errors like false negative and false positive are created by availability attacks, so that classifier is of no use. This type of violation also bans legitimate user's gains to it. Availability violation creates legitimate samples to being classified as malicious.

2.1.2.3 Privacy

In privacy violation, information is gained from the learner, by understanding the security of users of system, by the adversary.

2.1.3 Types of Attacks Specificity

2.1.3.1 Targeted

In targeted specificity, some definite samples are taken into account. That means target is on unique or small set of samples. For example, some definite spam emails being classified as legitimate. At a particular input, attacker may be targeted.

2.1.3.2 Indiscriminate

In indiscriminate specificity, target is on larger set of samples. It involves a very general class of points, that is any regular (false negative), since goal of indiscriminate specificity is more flexible. When input fails attacker may be indiscriminate.

2.2 Evaluation Methods Performance Limitations under Attack

In adversarial environments, performance evaluation methods [1] are k-fold cross validation, and bootstrapping. The method that is k-fold cross validation method is used for determining the system performance. In bootstrapping, the process is started and processes without any external input. The target of these methods is to find out work done by the pattern classifier when the operation is complete by using input data *D*. The data that occurs when operation is running uses same data distribution as that of *D*, on this assumption these methods are based. These methods again samples the input data *D* for creating the single or more pairs of training and testing that uses same distribution as that of data *D* [15]. The classification problem highly non-stationary, and makes it very difficult to estimate that how much and which types of attacks a classifier faces when operation is running, that is, how the data distribution will change, due to the presence of brilliant and adaptive adversary. Exploratory and causative attacks effects testing data which is processed by intelligent classifier. When the classifier retrains online, then causative attack effects on training data only [6], [8], [16]. When the classifier is under attack, then testing data may uses a distinct distribution than that of training data, during operation is run.

2.3 Reactive and Proactive Arms Race

"Reactive" arms race [1] between classifier designer and adversary occurs due to the security problems. Adversary understand the classifier defenses, and develops attacks scenarios to overthrown the attacks, at every step. The designer update the classifier, if required by understanding the attack samples, by checking it on advanced attack samples, and/or insert the characteristics that detects the attacks. For example, in spam filtering and malware detection, arms race can be examined, as a considerable increment in violation, attacks sophistication and also countermeasures.

Arms race can be described as, first, existing pattern recognition system is examined by the adversary, and data is manipulate to change the security of system. Some knowledge of the words is combines by spammer, which is used by targeted anti-spam filter for blocking spam and changes the spam email textual content. The designer of pattern recognition system reacts by updating the system and understanding the attack samples (see Figure 1 [1]). The next generation of security obligations is not anticipate by the "reactive" arms race, and the system remains vulnerable to advanced attacks, because they do not attempts to forecast future attacks. A "proactive" approach in which the designer should also attempt to anticipate the adversary's strategy (i) analyzing the relevant hazards, (ii) if required, design of countermeasures for the system, and (iii) before deploying the pattern recognition system, repeat the process for new design. The "proactive" arms race [1] shows the proactive version of arms race (see Figure 2 [1]). The target of evaluation of security is to address above issue (i) that is to replicate the no of attack samples that may be obtained when is runs and estimate the effect of targeted attacks on classifier, for highlighting the obligations. This equals to achieving what-if analysis [20] that is general practice in security evaluation. Also the above issue (ii) that is design of secure classifiers is a countermeasure, which is suggested by security evaluation that remains a problem.

2.4 Security by Design and Security by Obscurity

The most general scenario is used in cryptography and engineering is security by obscurity [1], for making a secure system. This system believes in the some personal information of the system to adversary. In security by design [1], the system is designed from ground up to be secure, beyond considering that some system may be find out by the adversary.

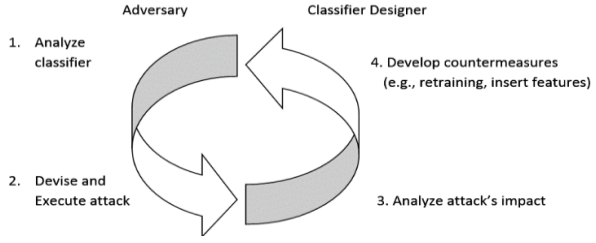

Fig 1: Classical representation of reactive arms race [1]

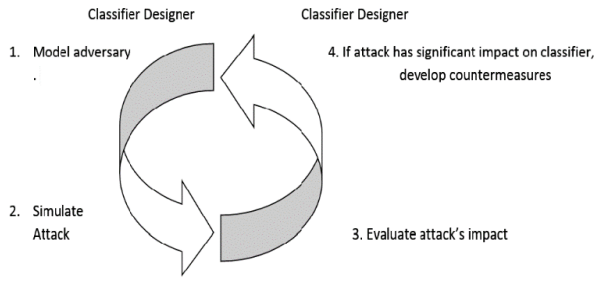

Fig 2: Classical representation of proactive arms race [1]

2.5 Some Previous Task On evaluation of Security

Following points shows the previous task on security evaluation.

2.5.1 Robust Linear Classifiers Design

In Robust linear classifiers design [2], to increase the robustness of linear classifiers with Boolean characteristics against changes in malicious samples whose target is to get them being classified as legitimate at operation stage is defined in [3]. In [3] defined that the robustness of classifiers is corresponds to the number of characteristics that modifies in malicious attack samples. A text classifier is skilled on Boolean characteristics, every one showing presence or absence of word in spam email, this occurs in spam filtering scenario. The robustness is evaluated in a sense that spammer can insert or delete the actual spam message. Adversary manipulates his samples by changing only those values of characteristics to evade the classifier, if some characteristics are very distinct on training samples. Adversary finds out the very distinct characteristics in scenarios like spam filtering and network intrusion detection system. By avoiding to over emphasis characteristics which are very distinct on training samples, robustness of classifier can be improved. This robustness is defined in averaging [3]. Averaging [3] method is based on multiple classifier system approach (MCS). Let, $f(x) = sign(g(x)) \in \{-1,+1\}$ is decision function for linear classifier, $x = x_1 \ldots x_n$ is n dimensional feature vector, linear discriminant function is $g(x) = \sum_{i=i}^{n} w_i x_i + w_0$, $(w_0 \cdots\cdots w_n)$ are features vectors, and the labels for legitimate, malicious samples are -1, +1 respectively. The discriminant function $g(x)$ is formed by averaging the ones of L distinct linear classifiers $g_1(x) \ldots\ldots g_n(x)$ and the linear classifier is generated with that discriminant function. By using, L distinct randomly selected subset of actual characteristics set, linear classifiers are generated, when chosen learning algorithms runs on same training samples. By setting the values of non-selected characteristics to zero, this is obtained in all training samples. The weights of $g(x)$ that is $(w_1 \cdots\cdot w_n)$ equal the average of corresponding L classifiers weights $w_i = \frac{1}{L}\sum_{j=1}^{L} w_i^j$. At training phase, the increment in computational cost, while the same individual linear classifier cost incurred at operation stage. This method also can be used to characteristics feature over fitting and under fitting, training set is not sufficient of representative of distribution of samples at operation stage [22].

2.5.2 Robust Linear Classifiers Design using Multiple Classifier System (MCS)

This method is defined in [2]. A more identical technique to the averaging method, because it shows two characteristics is called as Random Subspace method (RSM). First, in improving the classification accuracy with respect to the single classifier skilled with same learning algorithm, RSM is very effective. In adversarial classification system this characteristics makes good tradeoff between accuracy and robustness. Second, RSM corresponds to the group of popular MCS determination techniques, depends on randomization, whose another representatives are bagging [4] and Random forest method [5]. In adversarial classification system, this opens useful perfectives on potential usefulness of randomization-based MCS techniques.

2.5.2.1 Bagging

In randomization, MCS is constructed by giving training to the base classifier on distinct training sets, which is obtained by inserting some randomness to actual one. The bagging [4] is method of this kind. On bootstrap replications of the original training set, the individual classifiers are trained in bagging [4] method (see Algorithm 1 [2]).

Algorithm 1: Bagging [2]

Input: A set of N training samples $T = \{(x_i, y_i)\}$, where $i = 1 \ldots n$, learning algorithm L, ensemble size M.

Output: A classifier ensemble $\{f_1(x) \ldots\ldots f_M(x)\}$.

for $k = 1 \ldots\ldots M$ **do**

construct a bootstrap replication T_k by randomly drawing with replacement N samples from T.

end for

return $\{f_1(x) \ldots\ldots f_M(x)\}$.

2.5.2.2 Random Forest Method

The idea of training set resampling and random feature subset selection is combined in this method. For only decision trees random forest method [5] is applied.

2.5.3 Evaluation of Classifiers Security on Causative Attacks

In [6] defined that how to evaluate classifiers security under attacks that are causative. The adversary alters the training data with transformation A^{train} in the causative attacks. The various types of forces that attacker have that ranges from arbitrary control over some fraction of training instances to a biasing influence over the information. The learner causes it to produce a bad classifier, because attacker uses its various forces. Causative adversary uses A^{eval} to adjust the evaluation data in exploratory attacks. Typically, a causative adversary uses A^{train} and A^{eval} data for achieving his purpose. But, in some causative adversary coordinate with training data.

2.5.4 Evaluation of Classifiers Security under Class Imbalance Condition

The case when in a classification task, there are many more instances of some classes than others, is known as class imbalance condition. The classifiers in general perform poorly

because they tend to concentrate on the large classes and disregard the ones with few examples this is the problem in this condition. In [7] described evaluation of classifiers security at the class imbalance scenario. The adversary having the knowledge about encryption and decryption algorithms. The secret key sk shared between Alice and Bob is not known to adversary. The set of algorithms that is *F* running in good amount of time is the capability of adversary. Then (m_0, m_1) are any messages, *C* is ciphertext, adversary $A \in F$ cannot guess which message is encrypted with probability greater than 1/2. The adversary task is to construct an algorithm $A \in F$ which guess probability greater than 1/2.

3. A MODEL FOR PATTERN CLASSIFIERS SECURITY EVALUATION

A model for pattern classifiers security evaluation is described in following points.

3.1 Adversary's Model and Attack Scenario

Adversary's model and attack scenario is defined in [1]. Attack scenario is application specific issue. The designer of pattern recognition takes the help from attack scenario guidelines. The adversary acts rationally to attain a given goal, according to her knowledge of the classifier, and her capability of manipulating data, on this assumption this model is based.

3.1.1 Goal of the Adversary

The desired security violations such as integrity, availability or privacy and attacks specificity such as targeted, indiscriminate on which this functions based, as per taxonomy [8], [6]. To maximize the fraction of misclassified malicious samples [3], [6], [10] is the goal of indiscriminate violation. To obtain some specific, confidential information from the classifier by exploiting the class labels [6], [11], [12] is the goal of is the goal of targeted privacy violation. By minimizing the number of query samples that the adversary has to issue [6], [11], and [12] is the goal of privacy violation.

3.1.2 Knowledge of the Adversary

The knowledge can be described in terms of-

- the training data
- the feature set
- the decision function's type and learning algorithm
- the feedback available with the classifier

Note that, realistic and minimal assumptions about what can be kept fully private from the adversary should be done [6].

3.1.3 Capability of the Adversary

That means, the adversary has authority on training and testing data. This can be described in terms of-

- the influence of attack that can be causative or exploratory [6], [8]
- the class priors are affected by attacks to what extent
- the adversary controls which and how many training and testing samples in each class
- By taking into account only application-specific constraints (for example, malicious samples functionality cannot be compromised [3], [10], [13], [14]), which characteristics are manipulated and to what extent.

3.1.4 Strategy of the Attack

How training and testing data should be quantitatively changed to optimize the objective function characterizing the goal adversary, this defines the attack strategy. The changes are described in terms of-

- how the class priors are changed
- samples of each class is affected by the attack to what fraction
- how the characteristics are changed by attack

3.2 Data Distribution's Model

In adversarial classification, adversary creates samples at operation stage which is distinct from those in design stage, it causes that training data is not illustrative of testing data, since the problems in adversarial classification are non-stationary [3], [16]. Let, a problem of classifier design that consists of differentiate between legitimate (L) and malicious (M) samples. For this design, let $D = \{x_i, y_i\}$ is a set of n labelled samples has been collected, and a set of d features have been extracted, where $i = 1.....n$, x_i is d dimensional feature, and the class label is $y_i \in \{L, M\}$ that is it should be legitimate or malicious. After its deployment, when classifier is not under attack, then $p_{tr}(Y) = p_{ts}(Y) = p_D(Y) = p_{tr}(X|Y) = p_{ts}(X|Y) = p_D(X|Y)$. The components of p_{tr} and p_{ts} are not affected by the attack, the above assumption is applied to this component, by considering that they remains same related to distribution p_D [1].

When the distributions $p(x), p(X|Y)$ are under attack, then they easily defined depends on assumptions of attack strategy. Then class priors $p_{tr}(Y)$ and $p_{ts}(Y)$ are described on the basis of first assumption of attack strategy. Class conditional distributions $p_{tr}(X|Y), p_{ts}(X|Y)$ are described on the basis of second and third attack strategy. Then $p(X|Y)$ is a mixture controlled by Boolean random variable *A*, that shows sample subject to attack (*A*=*T*) or not (*A*=*F*).

$$p(X|Y) = p(X|Y, A=T)p(T|Y) + p(X|Y, A=F)p(F|Y)$$

The samples of the component $p(X|Y, A=T)$ is known to be attack samples, to maintain that their distribution is distinct from that of samples $p(X|Y, A=F)$. The stationary assumption holds; for the samples that are not affected by the attack.

$$p(X|Y, A=F) = p_D(X|Y) \quad \text{..... (I).}$$

According to the third assumption of attack strategy, the distribution is $p(X|Y, A=T)$. Similarly,

$p(X|Y, A=F)$ is defined. To, factorize $p(X, Y, A)$ consider the model (see Figure 3 [1]).

$p(X,Y,A) = p(Y)\,p(A|Y)\,p(X|Y,A)$ …… (II).

The distribution, $p(X,Y,A=F)$ changes over time, as per system $p(X,Y,A=F,t)$. By assuming this, classifiers security evaluation subject to temporal variations of data distribution. So, the classifier security at time *t* is evaluated by considering the distribution $p(X,Y,A=T,t)$ as an action of $p(X,Y,A=F,t)$ [1].

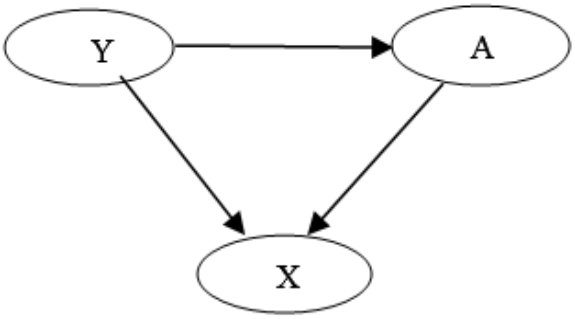

Fig 3: Model of p_{tr} and p_{ts} [1]

3.3 Generation of Training and Testing Set

Training (TR) and testing sets (TS) are constructed from the distributions $p_{tr}(X|Y)$ and $p_{ts}(X|Y)$ (see Algorithm 2 [1]). By using resampling techniques, such as cross validation and bootstrapping, training and testing sets (D^i_{TR}, D^i_{TS}) are obtained from data set *D*. To, generate a training set TR^i, how to modify the sets D^i_{TR} is described that uses the distribution $p_{tr}(X,Y)$. For the simplicity, the superscript i is omitted. Then training set TR^i is generated from D^i_{TS}. When the performance of classifier is trained on TR^i and tested on TS^i is averaged then security evaluation is carried out by classical method [1].

Algorithm 2: Generation of training and testing set [1]

Input: The distributions $p(Y)$ and $p(A|Y)$; n that is number of samples if the distributions $p(X|Y=y,A=a)$ for $y \in \{L,M\}$, $a \in \{T,F\}$, defined analytically, or the set of samples $D^{y,a}$, else.

Output: A set S that is (*TR* or *TS*) drawn from $p(Y)\,p(A|Y)\,p(X|Y,A)$

$S \leftarrow \phi$

for $i = 1....n$ **do**

sample y from $p(Y)$

sample a from $p(A|Y=y)$

draw a sample *x* from $p(X|Y=y,A=a)$, if defined analytically, else sample with replacement from $D^{y,a}$.

$S \leftarrow S \cup \{(x,y)\}$

end for

return S

If training samples are not affected by the attack then $p_{tr}(X|Y) = p_D(X|Y)$, that is D_{TR} equal to TR. Else, two solutions are-(i) TR is generated by sampling the model $p(X,Y,A)$,for each $Y \in \{L,M\}$, if $p_{tr}(X|Y,A)$ is described. (ii) The distribution $D^{y,a}_{TR}$ is approximated as distribution of $D^{y,a}_{TR}$, if $p_{tr}(X|Y=y,A=a)$ is not described for *y* and *a*. If $p_{tr}(X|Y=y,A=a)$ is not defined analytically, then $D^{y,a}_{TR}$ is also generated [1].

The one and same distribution used for $D^{y,a}_{TR}$. In D_{TR}, the two distributions $D^{L,F}_{TR}$ and $D^{y,a}_{TR}$ sets equal to legitimate and malicious samples. So, the distribution is considered as $p_{tr}(X|Y=L,A=F)$ and $p_{tr}(X|Y=M,A=F): D^{L,F}_{TR} = \{(x,y) \in D_{TR} : y = L\}$, $D^{M,F}_{TR} = \{(x,y) \in D_{TR} : y = M\}$ Instead of $p_{tr}(X|Y=y,A=T)$, two sets of sample $D^{y,T}_{TR}$ must come, whereas $y = \{L,M\}$ as per third assumption of attack strategy [1].

4. ACKNOWLEGMENTS

The authors are highly indebted to the authors of various research papers that are helpful for preparing this survey paper. The authors are also grateful to SGGSIE&T, nanded for their support for completing this survey paper.

5. CONCLUSION

This paper gives information about the pattern classifiers security evaluation under attack that is in adversarial environments. To simulate realistic attack scenarios by giving practical guidelines, the described system defines adversary model, in terms of capability, knowledge, and goal. The distribution of training and testing samples is affected by the targeted attacks, the described system also defines data distribution's model. The defined model is data dependent, because security evaluation is carried out empirically. The model provides high level guidelines, because it is not application specific.

Pattern classification consist of data pre-processing, feature extraction, model selection, classifier training, and classification that are helpful for security evaluation. The described model helps for the systems that faces attack problems during their normal operation. The future work includes finding the solution for various application scenarios with respect to different attacks.

6. REFERENCES

[1] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of Pattern Classifiers under Attack," IEEE Transactions on Knowledge and Data Engineering, vol. 26, No. 4, April 2014. Available online at: https://pralab.diee.unica.it/sites/default/files/biggio13-tkde.pdf

[2] B. Biggio, G. Fumera, and F. Roli, "Multiple Classifier Systems for Robust Classifier Design in Adversarial Environment," Int'l J. Machine Learning and Cybernetics, vol. 1, no. 1, pp. 27-41, 2010.

[3] A. Kolcz and C. H. Teo, "Feature weighting for improved classifier robustness," In 6th Conf. on Email and Anti-Spam (CEAS), 2009.

[4] L. Breiman. Bagging predictors. Machine Learning, 24(2):123–140, 1996

[5] L. Breiman. Random forests. Machine Learning, 45:5–32, 2001.

[6] 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.

[7] A. A. Cardenas and J. S. Baras, "Evaluation of Classifiers: Practical Considerations for Security Applications," Proc. AAAI Workshop Evaluation Methods for Machine Learning, 2006.

[8] 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.

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

[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] D. Lowd and C. Meek, "Adversarial Learning," Proc. 11th ACMSIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 641-647, 2005.

[12] A. Adler, "Vulnerabilities in Biometric Encryption Systems," Proc.Fifth Int'l Conf. Audio- and Video-Based Biometric Person Authentication, pp. 1100-1109, 2005.

[13] 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. 196-179, 2009.

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

[15] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification. Wiley-Interscience Publication, 2000.

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

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

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

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

[20] S. Rizzi, "What-If Analysis," Encyclopedia of Database Systems, pp. 3525-3529, Springer, 2009.

[21] 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.

[22] C. Sutton, M. Sindelar, and A. McCallum. Feature bagging: Preventing weight undertraining in structured discriminative learning. IR 402, University of Massachusetts, 2005.