Applications of Machine Learning in Cyber Security

Vitaly Ford and Ambareen Siraj
Computer Science Department, Tennessee Tech University
Cookeville, TN, 38505, USA
vford42@students.tntech.edu, asiraj@tntech.edu

Abstract

Machine learning techniques have been applied in many areas of science due to their unique properties like adaptability, scalability, and potential to rapidly adjust to new and unknown challenges. Cyber security is a fast-growing field demanding a great deal of attention because of remarkable progresses in social networks, cloud and web technologies, online banking, mobile environment, smart grid, etc. Diverse machine learning methods have been successfully deployed to address such wide-ranging problems in computer security. This paper discusses and highlights different applications of machine learning in cyber security. This study covers phishing detection, network intrusion detection, testing security properties of protocols, authentication with keystroke dynamics, cryptography, human interaction proofs, spam detection in social network, smart meter energy consumption profiling, and issues in security of machine learning techniques itself.

keywords: Security, machine learning, survey.

1 Introduction

Alongside of fast evolvement of web and mobile technologies, attack techniques are also becoming more and more sophisticated in penetrating systems and evading generic signature-based approaches. Machine learning techniques offer potential solutions that can be employed for resolving such challenging and complex situations due to their ability to adapt quickly to new and unknown circumstances. Diverse machine learning methods have been successfully deployed to address wide-ranging problems in computer and information security. This paper discusses and highlights different applications of machine learning in cyber security.

The paper is structured as follows. Section 2 describes various applications of machine learning in information security: phishing detection, network intrusion detection, testing security properties of protocols, authentication with keystroke dynamics, cryptography, human interaction proofs, spam detection in social network, smart meter energy consumption profiling, and issues in security of machine learning techniques itself. Section 3 concludes with future work.

2 Methodology

2.1 Phishing Detection

Phishing is aimed at stealing personal sensitive information. Researchers [2] have identified three principal groups of anti-phishing methods: detective (monitoring, content filtering, anti-spam), preventive (authentication, patch and change management), and corrective (site takedown, forensics) ones. These categories are summarized in Table 1.

Table 1: Phishing and Fraud Solutions [1, 2]

Detective Solutions	Preventive Solutions	Corrective Solutions
1. Monitors account life cycle 2. Brand monitoring 3. Disables web duplication 4. Performs content filtering 5. Anti-Malware 6. Anti-Spam	1. Authentication 2. Patch and change management 3. Email authentication 4. Web application security	1. Phishing site takedown 2. Forensics and investigation

A comparison of phishing detection techniques appears in [1]. It was observed that many phishing detection solutions under consideration have a high rate of missed detection. Researchers compared six machine learning classifiers, using 1,171 raw phishing emails and 1,718 legitimate emails, – "Logistic Regression (LR), Classification and Regression Trees (CART), Bayesian Additive Regression Trees (BART), Support Vector Machines (SVM), Random Forests (RF), and Neural Networks (NNets)". The error rates of all the above-mentioned classifiers are summarized in Figure 1.

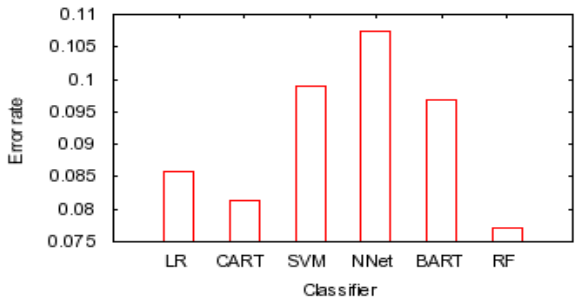

Figure 1: The error rates of classifiers [1]

For experimentation, text indexing techniques were used for parsing the emails. All attachments were removed, "header information of all emails and html tags" from the emails' bodies as well as their specific elements were extracted. Afterwards, a stemming algorithm was applied and all the irrelevant words were removed. Finally, all items were sorted according to their frequency in emails. As a result of this work, it can be concluded that LR is a more preferable option among users due to low false positive rate (usually, users would not want their legitimate emails to be misclassified as junk). Also, LR has the highest precision and relatively high recall in comparison with other classifiers under contemplation. The comparison of precision, recall, and F-measure is given in Table 2.

Table 2: Comparison of precision, recall, and F1 [1]

Classifier	Precision	Recall	F1
LR	95.11 %	82.96 %	88.59%
CART	92.32 %	87.07 %	89.59 %
SVM	92.08 %	82.74 %	87.07 %
NNet	94.15 %	78.28 %	85.45 %
BART	94.18 %	81.08 %	87.09 %
RF	91.71 %	88.88 %	90.24 %

Zhuang et al. [6] developed an automatic system for phishing detection applying a cluster ensemble of several clustering solutions. A feature selection algorithm for extracting various phishing email traits was used, which was: Hierarchical Clustering (HC) Algorithm that adopted cosine similarity (using the TF-IDF metric) for measuring the similarity between two points, and K-Medoids (KM) Clustering approach. The proposed methods for phishing website and malware categorization have about 85% performance. The architecture of their Automatic Categorization System (ACS) is shown in Figure 3.

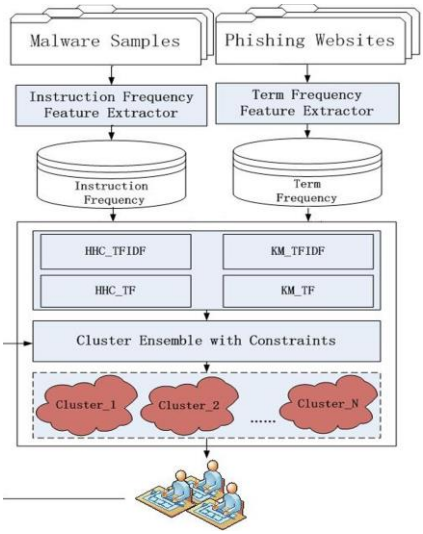

Figure 3: The Architecture of ACS [6]

First, the ACS parses the malware samples and phishing web-sites. It extracts terms and specific malware instructions and saves them to a database. After that the system applies the information retrieval algorithm for calculating the TF-IDF metrics. Then, the ACS utilizes the ensemble of clustering algorithms and, taking account of constrains manually generated by security experts, splits the data into clusters.

2.2 Network Intrusion Detection

Network Intrusion Detection (NID) systems are used to identify malicious network activity leading to confidentiality, integrity, or availability violation of the systems in a network. Many intrusion detection systems are specifically based on machine learning techniques due to their adaptability to new and unknown attacks.

Lu et al. [8] proposed a unified effective solution for improving Genetic Network Programming (GNP) for misuse and anomaly detection. Matching degree and genetic algorithm were fused so that redundant rules can be pruned and efficient ones can be filtered. The system was tested on KDDcup99 [22] data to demonstrate its efficiency. The proposed pruning algorithm does not require "prior knowledge from experience". The rule is pruned if the average matching degree is less than some threshold. On the training step, 8,068 randomly chosen connections were fed into their system (4,116 were normal, 3,952 – smurf and neptune attacks). After training the system, the proposed solution was tested on 4,068 normal connections and 4,000 intrusion connections. The accuracy (ACC) is reported to be 94.91%, false positive rate (FP) is 2.01%, and false negative rate (FN) is 2.05%. Table 4 displays the performance comparison of different algorithms including the proposed one.

Table 4: The performance comparison of NID systems [8]

NID	Detection Rate	ACC	FP	FN
Unified detection (w/ two-stage rule pruning)	97.75%	94.91%	2.01%	2.05%
Unified detection (w/o two-stage rule pruning)	95.79%	90.17%	4.41%	3.75%
GNP-based anomaly detection	86.89%	–––	18.4%	0.75%
GNP-based misuse detection	94.71%	–––	3.95%	8.54%
Genetic programming	90.83%	–––	0.68%	–––
Decision trees	–––	89.70%	–––	–––
Support vector machines	95.5%	–––	1.0%	–––

Subbulakshmi et al. [9] developed an Alert Classification System using Neural Networks (NNs) and Support Vector Machines (SVM) against Distributed Denial of Service (DDoS) attacks. For simulating a real DDoS attack, a virtual environment was used with "Snort" tool for intrusion detection, and "packit" for generating network packets and sending them to the target machine. The alerts generated by the snort intrusion detection tool were captured and fed into a back-propagation neural network and support vector machines for classifying the alerts as true-positives or false-positives. The researchers claimed that this process reduced the total number of alerts to process by 95%. The average accuracy of neural network alert classification is 83% whereas for support vector machines, it is 99%. A comparison of NNs and SVM with the Threshold Based Method (TBM) and Fuzzy Inference System (FIS) is shown in Table 5.

Table 5: The Comparison of NNs, SVM, TBM, and FIS [9]

Type of attack	Classification Accuracy			
	TBM	FIS	NNs	SVM
UDP	75.00 %	84.30 %	85.22 %	99.28 %
TCP SYN	73.00 %	82.34 %	83.56 %	99.45 %
ICMP	73.45 %	81.24 %	83.21 %	99.39 %
ICMP SMURF	70.14 %	77.89 %	81.27 %	98.40 %

Sedjelmaci and Feham [10] propose a hybrid solution for detecting intrusions in a Wireless Sensor Network (WSN). A clustering technique is employed for reducing the amount of information to process and the energy to consume. In addition, Support Vector Machines (SVMs) with misuse detection techniques are used for identifying network anomalies. The system consists of many distributed intrusion detection nodes that communicate with each other to identify attacks. The efficient algorithm for choosing optimal distributed SVMs is shown in Figure 4. Denial of Service and Probe attacks were considered for testing which are most common in WSN environment than any other ones. The performance evaluation of the proposed distributed system is displayed in Table 6.

Figure 4: Optimal Distributed SVMs Selection Process [10]

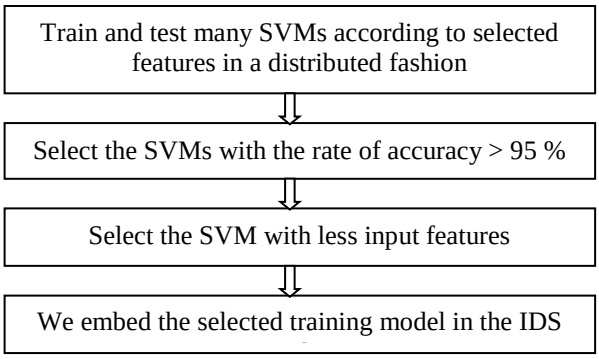

Table 6: Performance Evaluation of the Distributed IDS [10]

Number of Features	Accuracy	Detection Rate
9	97.80 %	93.66 %
7	98.47 %	95.61 %
5	96.95 %	91.21 %
4	98.39 %	95.37 %

In comparison with a centralized intrusion detection system [11], the proposed solution obtains a higher accuracy when there is not enough training data (the accuracy rate is 98%). Also, the proposed approach claims to reduce energy consumption.

2.3 Authentication with Keystroke Dynamics

Revett et al. [12] proposed applying a Probabilistic Neural Network (PNN) for keystroke dynamics. Generally, keystroke dynamics represents "a class of behavioral biometrics that captures the typing style of a user". The system was evaluated on a dataset containing login/password keystrokes of 50 people. Revett et al. asked 30 of them to login as imposters multiple times instead of legitimate users. Eight different attributes were monitored during enrollment and authentication attempts. These attributes were: digraphs (DG, two-letter combinations), trigraphs (TG, three-letter combinations), total username time, total password time, total entry time, scan code, speed, and edit distance. Subsequently, the data was fed into the PNN system and tested. The accuracy of classification of legitimate/imposter equaled 90%. Also, PNN was compared to a multi-layer perceptron neural network (MLPNN) with back-propagation and it was found that PNN training time is 4 times less than MLPNN one. The summation of False Acceptance and False Rejection Rates of PNN is 1.5 times less than MLPNN one. The comparison of the MLPNN and PNN algorithms can be seen in Table 7. The values of this table are the summation of the False Acceptance Rate (FAR) and False Recognition Rate (FRR).

Table 7: FAR + FRR of PNN and MLPNN [12]

Attributes	PNN, %	MLPNN, %
All	3.9	5.7
Primary only	5.2	6.5
Derived only	4.2	6.2
DG + primary	4.4	5.3
TG + primary	4.0	5.8
Edit distance only	3.7	5.0

2.4 Testing Security of Protocol Implementation

Shu and Lee [13] a new notion of applying machine learning for "testing security of protocol implementation". The researchers mainly focused their research on "Message Confidentiality (secrecy) under Dolev-Yao model of attackers" that tries to inject a message to the original one [14]. Generally, there is no comprehensive solution for a holistic testing of a protocol implementation security. However, experiments can be fulfilled with respect to a problem restricted to a finite number of messages. And the main goal of their paper is to find some weak spots (that violate security) in a protocol black-box implementation, deploying L* learning algorithm [15]. In this algorithm the researchers created a teacher that performs three principal actions: 1) Generating an output query given an input sequence; 2) Generating a counterexample that a system outputs as an incorrect result when analyzing it; 3) Augmenting the alphabet, appending new input symbols in addition to the existing ones. They showed the effectiveness of their proposed technique on testing three real protocols: Needham-Schroeder-Lowe (N-S-L) mutual authentication protocol, TMN key exchange protocol, and SSL 3.0 handshake protocol. As a result, their system identified the introduced flaws in N-S-L and TMN. Also, it confirmed that SSL is secured.

2.5 Breaking Human Interaction Proofs (CAPTCHAs)

Chellapilla and Simard [16] discuss how the Human Interaction Proofs (or CAPTCHAs) can be broken by utilizing machine learning. The researchers experimented with seven various HIPs and learned their common strengths and weaknesses. The proposed approach is aimed at locating the characters (segmentation step) and employing neural network [17] for character recognition. Six experiments were conducted with EZ-Gimpy/Yahoo, Yahoo v2, mailblocks, register, ticketmaster, and Google HIPs. Each experiment was split into two parts: (a) recognition (1,600 HIPs for training, 200 for validation, and 200 for testing) and (b) segmentation (500 HIPs for testing segmentation). On the recognition stage, different computer vision techniques like converting to grayscale, thresholding to black and white, dilating and eroding, and selecting large CCs with sizes close to HIP char sizes were applied. Figure 5 demonstrates some of those algorithms in operation.

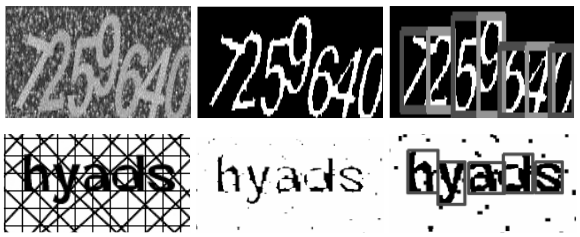

Figure 5: Examples of segmentation [16]

The following Table 8 compiles the experimentation results:

Table 8: Success Rates for Segmentation and Recognition steps

HIP	Success rate for segmentation	Success rate for recognition given correct segmentation	Total
Mailblocks	88.8 %	95.9 %	66.2 %
Register	95.4 %	87.1 %	47.8 %
Yahoo/EZ-Gimpy	56.2 %	90.3 %	34.4 %
Ticketmaster	16.6 %	82.3 %	4.9 %
Yahoo ver. 2	58.4 %	95.2 %	45.7 %
Google/Gmail	10.2 %	89.3 %	4.89 %

It was reported that the segmentation stage is relatively difficult for the following reasons: (a) computationally expensive; (b) complex segmentation function because of an immense non-valid pattern space; and (c) difficulty in identification of valid characters.

2.6 Cryptography

Yu and Cao [18] developed a fast and efficient cryptographic system based on delayed chaotic Hopfield neural networks. The researchers claim that the proposed system is secured due to "the difficult synchronization of chaotic neural networks with time varying delay".

Kinzel and Kanter [20] show how two synchronized neural networks can be used for a secret key exchange over a public channel. Basically, on the training stage two neural networks start with random weight vectors and receive an arbitrary identical input sequence every cycle. The weights are changed only if the outputs of both neural networks are the same. And after a short period of time the corresponding weight vectors of both neural networks become identical. The researchers have demonstrated that it is computationally infeasible to perform some attacks.

2.7 Social Network Spam Detection

K. Lee et al. [7] observed that spammers exploit social systems for employing phishing attacks, disseminating malware, and promoting affiliate websites. For protecting social systems against those attacks, a social honeypot was developed for detecting spammers in social networks like Twitter and Facebook. The proposed solution is based on Support Vector Machine (SVM) and has a high precision as well as low false positive rate. A social honeypot represents a legitimate user profile and a corresponding bot, which gathers both legitimate and spam profiles and feeds them into the SVM classifier. For evaluating the

performance of the proposed machine learning system, the researchers examined MySpace and Twitter networks. Various legitimate user accounts were created in both social networks and data were collected over several months. Deceptive spam profiles, like click traps, friend infiltrators, duplicate spammers, promoters, and phishers were manually singled out into several groups. The SVM was fed with the data (for MySpace: 388 legitimate profiles and 627 deceptive spam profiles; for Twitter: 104 legitimate profiles, 61 spammers' and 107 promoters' profiles. Results demonstrate spam precision to be 70% for MySpace and 82% for Twitter.

2.8 Smart Meter Data Profiling

In our recent work, we have applied fuzzy c-means clustering for smart meter data profiling [24]. Our research demonstrates that by having access to energy consumption traces captured by smart meters, one can implement a disaggregation technique for deducing consumer energy consumption profiles, which can compromise privacy of consumers and have the potential to be used in undesirable ways. Time frame between when the customer leaves and returns home offers opportunities for home invasion, marketing by phone, or even children behavior profiling.

For instance, our analysis of a three-day data sequence for a smart meter (Figure 6) reveals certain pattern of energy consumption behavior. Here axis *X* denotes date/time of the measurement, and axis *Y* denotes energy consumption value in kW/h. From these observations, it can be inferred that the consumer is a service providing business (like a store/eatery) rather than a household as the energy consumption is at its peak consistently between 8:30 A.M. until 10:00 P.M. (Figure 7). It can further be inferred that it is using certain types of appliances consuming 0.55 kW/h during a nighttime period every half an hour. It is likely that these appliances would be security and/or fire detecting devices with periodic small and persistent energy consumption.

Figure 8 represents another pattern observed for a single customer randomly chosen from the dataset. It shows that the value of the consumed energy varies from 0 to 0.1 kW/h between 1 A.M. to 8 A.M. Therefore, it can be inferred that during this time period, the customer does not usually use any appliances. This maybe because: 1) if it is a residential home, the customer sleeps at that time; 2) if it is a business, it is not active at that period of time. Taking into consideration the fact that the customer usually consumes from 0.358 to 0.548 kW/h during 8 P.M. – 12 A.M., it can be deduced that we are looking at a typical working household (where people sleep at night, go to work all day and come back to have dinner, watch TV and then go to bed again).

Also, by having access to detailed energy consumption data, one can infer information about appliances usage and spammers can exploit such information for their own benefit. On the other side, utility corporations can make use of such knowledge to detect abrupt changes in consumer usage patterns, which can be used to detect energy fraud – an important issue in the smart grid.

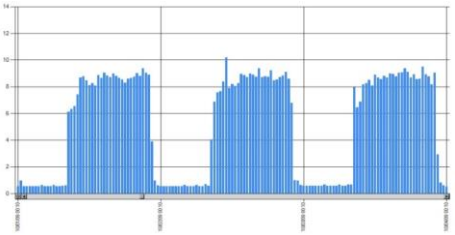

Figure 6: Energy Consumption Profile for One Smart Meter for Three Consecutive days [24]

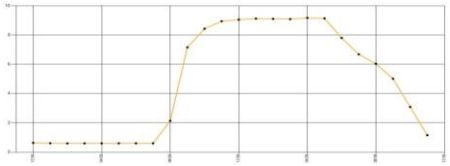

Figure 7: Mean Energy Consumption per Half an Hour [24]

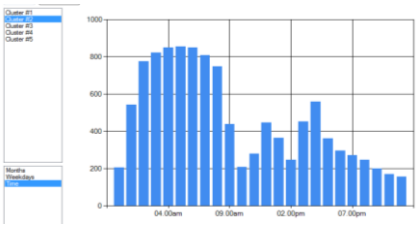

Figure 8: Energy Consumption Profile for Single Customer [24]

2.9 Security of Machine Learning

M. Bareno et al. [21] discuss many diverse ways for compromising machine learning system. The researchers provides a comprehensive taxonomy of different attacks aimed at exploiting machine learning systems: (a) *Causative* attacks altering the training process; (b) Attacks on *integrity* and *availability*, making false positives as a breach into a system; (c) *Exploratory* attacks exploiting the existing vulnerabilities; (d) *Targeted* attacks directed to a certain input; (e) *Indiscriminate* attacks in which inputs fail.

The researchers proposed the Reject On Negative Impact (RONI) defense. RONI ignores all the training data points that have a substantial negative impact on the classification accuracy.

There are two main types of defenses they discussed. First type is a defense against exploratory attacks, in which an attacker can create an evaluation distribution that the learner predicts poorly. For defending against this attack,

the defender can limit the access to the training procedure and data, making it harder for an attacker to apply reverse engineering. Also, the more complicated a hypothesis space is, the harder for an attacker to infer the learned hypothesis. In addition, a defender can limit the feedback (or send the deceitful one) given to an attacker so that it becomes harder to break into the system.

Second type is a defense against causative attacks, in which an attacker can manipulate both training and evaluation distributions. In this scenario, the defender can deploy the RONI defense in which the system has two classifiers. One classifier is trained using a base training set; another is trained with not only a base set but also the candidate instance. If the errors of those two classifiers significantly differ from each other, the candidate instance is treated as a malicious one.

As an example of applying the defensive RONI algorithm, the researchers simulated attacking the SpamBayes spam detection system [23] and showed the effectiveness of the system against Indiscriminate Causative Availability attacks.

3 Conclusion

Machine learning is an effective tool that can be employed in many areas of information security. There exist some robust anti-phishing algorithms and network intrusion detection systems. Machine learning can be successfully used for developing authentication systems, evaluating the protocol implementation, assessing the security of human interaction proofs, smart meter data profiling, etc. Although machine learning facilitates keeping various systems safe, the machine learning classifiers themselves are vulnerable to malicious attacks. There has been some work directed to improving the effectiveness of machine learning algorithms and protecting them from diverse attacks. There are many opportunities in information security to apply machine learning to address various challenges in such complex domain. Spam detection, virus detection, and surveillance camera robbery detection are only some examples.

References

[1] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A Comparison of Machine Learning Techniques for Phishing Detection", *APWG eCrime Researchers Summit*, October 4-5, 2007, Pittsburg, PA.

[2] Anti-Phishing Working Group, "Phishing and Fraud solutions". [Online]. Available: http://www.antiphishing.org/. [Accesses: April 4, 2013].

[3] M. Wu, R. C. Miller, and S. L. Garnkel, "Do security toolbars actually prevent phishing attacks?" in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006.

[4] L. F. Cranor, S. Egelman, J. Hong, and Y. Zhang, "Phinding phish: An evaluation of anti-phishing toolbars", *Technical Report CMU-CyLab-06-018*, CMU, November 2006.

[5] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya, "Phishing email detection based on structural properties", *in NYS Cyber Security Conference*, 2006.

[6] W. Zhuang, Y. Ye, Y. Chen, and T. Li, "Ensemble Clustering for Internet Security Applications", in *IEEE xplore*, December 17, 2012.

[7] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots + machine learning", *SIGIR'10*, July 19-23, 2010, Geneva, Switzerland.

[8] N. Lu, S. Mabu, T. Wang, and K. Hirasawa, "An Efficient Class Association Rule-Pruning Method for Unified Intrusion Detection System using Genetic Algorithm", in *IEEJ Transactions on Electrical and Electronic Engineering*, Vol. 8, Issue 2, pp. 164 – 172, January 2, 2013.

[9] T. Subbulakshmi, S. M. Shalinie, and A. Ramamoorthi, *"Detection and Classification of DDoS Attacks using Machine Learning Algorithms"*, European Journal of Scientific Research, ISSN 1450-216X, Volume 47, No. 3, pp. 334 – 346, 2010.

[10] H. Sedjelmaci, and M. Feham, *"Novel Hybrid Intrusion Detection System for Clustered Wireless Sensor Network"*, International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.4, July 2011.

[11] T. H. Hai, E. N. Huh and M. Jo, *"A Lightweight Intrusion Detection Framework for Wireless Sensor Networks"*, Wireless Communications and mobile computing, Vol.10, Issue 4, pp. 559-572, 2010.

[12] K. Revett et al., *"A machine learning approach to keystroke dynamics based user authentication"*, International Journal of Electronic Security and Digital Forensics, Vol. 1, No. 1, 2007.

[13] G. Shu and D. Lee, "Testing Security Properties of Protocol Implementations – a Machine Learning Based Approach", in *Proceedings of 27th International Conference on Distributed Computing Systems (ICDCS'07)*, 2007.

[14] D. Dolev and A. Yao, "On the security of public-key protocols", *IEEE Transaction on Information Theory 29*, pages 198-208, 1983.

[15] D. Angulin, "Learning regular sets from queries and counterexamples", Information and Computation, 75, pp. 87-106, 1987.

[16] K. Chellapilla and P. Y. Simard, "Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)", in *Advances in Neural Information Processing Systems 17*, pp. 265-272, 2005.

[17] Simard PY, Steinkraus D, and Platt J, (2003) "Best Practice for Convolutional Neural Networks Applied to Visual Document Analysis," in *International Conference on Document Analysis and Recognition(ICDAR)*, pp. 958-962, IEEE Computer Society, Los Alamitos.

[18] W. Yu and J. Cao, "Cryptography based on delayed chaotic neural networks", Physics Letters A, Vol. 356, Issues 4–5, pp. 333-338, ISSN 0375-9601, August 14, 2006.

[19] J. Yang et al., "Cryptanalysis of a cryptographic scheme based on delayed chaotic neural networks", Chaos, Solitons & Fractals, Vol. 40, Issue 2, pp. 821-825, ISSN 0960-0779, April 30, 2009.

[20] W. Kinzel and I. Kanter, "Neural Cryptography", in *Proceedings of the 9th International Conference on Neural Information Processing*, Vol. 3, pp. 1351-1354, November 18-22, 2002.

[21] M. Barreno et al., *"The security of machine learning"*, Journal Machine Learning, Vol. 81, Issue 2, pp. 121-148, November 2010.

[22] Knowledge Discovery and Data Mining group, "KDD cup 1999". [Online]. Available: http://www.kdd.org/kddcup/index.php. [Accessed: March 3, 2013].

[23] SpamBayes Project Group, "SpamBayes". [Online]. Available: http://spambayes.sourceforge.net/. [Accessed: February 15, 2013].

[24] V. Ford and A. Siraj, "Clustering of smart meter data for disaggregation", in *Proceedings of IEEE Global Conference on Signal and Information Processing*, December, 2013.