J Ambient Intell Human Comput (2013) 4:235–247
DOI 10.1007/s12652-011-0073-z

ORIGINAL RESEARCH

Machine learning and soft computing for ICT security: an overview of current trends

Francesco Camastra · Angelo Ciaramella · Antonino Staiano

Received: 23 May 2011 / Accepted: 4 October 2011 / Published online: 20 October 2011

Abstract In the last years, people have been seeing the pervasive use of computer, communication technology and Internet, e.g., e-mail, online shopping, banking, gaming, Internet telephony, streaming. Unfortunately, the reliability of the Internet and its services, and in general *Information and Communication Technology* (ICT) devices, is undermined by insecurity issues. On the other hand, machine learning and soft computing techniques have been widely applied to disparate fields, becoming, in several cases, the leading technology. The aim of the work is to investigate the trends of the machine learning (ML) and soft computing (SC) methodologies for ICT security. In particular, it overviews ML and SC applications for three hot topics in ICT security: password-based schemes for access control, intrusion detection and spam filtering.

Keywords Machine Learning · Soft computing · ICT security · Password-based schemes for access control · Intrusion detection system · Spam filtering

1 Introduction

Computer security is one of the most important issues in the *Information and Communication Technology* (ICT) society. Nowadays, the Internet has become a universal communication platform representing the best expression of the *ubiquitous* technology envisioned by Weiser (1991) in the late 1980s. Internauts all over the world are connected either for work or personal use. E-mails have, by now, supplanted envelops and stamps. Everyone browses the latest news on his mobile or manages his money and personal data through a netbook seated down at a restaurant offering a free Wi-Fi. A main reason for the growth of this pervasive use of ICT has been the adoption of commercial services paving the way to e-commerce and multimedia services. Every company, university, governmental organization, critical plants (e.g., power plants) are globally connected providing their services through the Internet. Unfortunately, the reliability in the Internet and its services is undermined by network attacks. Personal as well as business computer systems are generally at risk to be remotely compromised and misused for illegal purposes. Today, a plethora of attacks plagues computers connected to the Internet, e.g., computer worms, malwares and trojan horses. Proliferation of these threats is driven by a criminal economy that rests on "business models" such as gathering of confidential data, disruption of services or distribution of spam messages. While a private citizen may receive very limited damages if he/she is a cyber attack victim, this becomes a serious threat to companies and governmental organizations. The WHID 2007 report[1] by the Web Application Security Consortium revealed that 67% of attacks in 2007 were profit motivated. There are many examples in recent news of cyber attacks. For instance, in 2009 it was revealed that the US power grid had been infiltrated by an intruder, leaving malware that was capable of shutting down the entire grid. In the same year, a major

F. Camastra · A. Ciaramella (✉) · A. Staiano
Department of Applied Science, University of Naples Parthenope, Centro Direzionale Isola C4, 80143 Naples, Italy
e-mail: angelo.ciaramella@ieee.org

F. Camastra
e-mail: camastra@ieee.org

A. Staiano
e-mail: antonino.staiano@uniparthenope.it

[1] WHID (Web-Hacking-Incident-Database) 2007 report is available on http://www.webappsec.org/projects/whid/statistics.shtml.

spy network (GhostNet) had been infiltrated more than 1,000 computers around the world, with victims such as foreign ministers and embassies.

Spamming is another example of cyber attack, a user receives unsolicited mail, ranging from more or less obvious commercial to phishing messages specifically designed to resemble legitimate e-mails. Spam mines the reliability of e-mails (Hoanca 2006) and causes so large economical impact that some countries made a legislation to ban spamming (Talbot 2008). It is clear that cyber attacks can threaten national security, prompting USA to open a *Cyber Security Office*[2] in the White House in 2009 followed shortly by the UK where the *Cyber Security Operations Centre*[3] was founded.

There are several mechanisms that can be adopted to increase the security in computer systems: *attack prevention* (firewalls, user names and passwords, and user rights), *attack avoidance* (encryption) and *attack detection* (intrusion detection systems). As witnessed by a plethora of security literature, machine learning (ML) (Bishop 2006) and soft computing (SC) (Zadeh 1994) methodologies offer the flexibility required to realize efficient tools for computer security and have the generalization capability necessary to address a wide range of security issues. This is why ML and SC techniques have been widely adopted in Computer Security and Computer Forensics (Stahl et al. 2010). ML models are often used to address *supervised learning* problems, in which a ML model is trained on a proper set of data, called *training set* along with their corresponding label, called *target*, or *unsupervised learning* problems, in which the ML model is trained on the training set alone, namely unlabeled (Bishop 2006). In these two categories of problems fall the most ML and SC methods applied to ICT security applications.

This work aims to provide the trends of the ML and SC methodologies for ICT security. ML and SC applications for three hot topics in ICT security, e.g., password-based schemes for access control, intrusion detection and spam filtering (SF), are overviewed.

The work is organized as follows: Sect. 2 is devoted to the password-based schemes for access control and in particular for password authentication, password strength, proactive password checking and password keystroke dynamics; in Sect. 3 intrusion detection systems are introduced and a taxonomy of the different applied ML and SC paradigms is provided; in Sect. 4 some aspects of the spam filtering are discussed; finally, in Sect. 5 conclusions are drawn.

[2] http://news.bbc.co.uk/2/hi/americas/8073654.stm.

[3] http://news.bbc.co.uk/2/hi/uk_news/politics/8118348.stm.

2 Password-based schemes for access control

A big challenge for computer scientists is the design of secure protocols for protecting partially shared or private resources. In computer security, access control provides the essential services as accounting, authentication and authorization (AAA). Even though several suitable techniques have been proposed in literature over the years (e.g., biometric identification schemes, smart cards) password-based schemes are still frequently used due to their simplicity.

In this section it is provided a survey on the ML and SC techniques recently proposed in the fields of password authentication, password strength, proactive password checking and password keystroke.

2.1 Password authentication

Password authentication is one of the mechanisms that is widely used to authorize a legitimate user (e.g., access to accounts, PINs, files, and so on) and that does not require the use of more expensive devices. Typically a keypad is required to record and save in a table on a non-volatile storage a (*user ID, password*) combination. Successively, this information is used to authenticate an user. The vulnerability of this system consists in the possible access of the password information table by an intruder. Moreover, insecurity can be limited by encrypting the (*user ID, password*) pairs prior to saving them in storage (Sibai et al. 2009). In the encrypted case the verification table need not be kept secured, because an intruder cannot decipher the original passwords from what is stored in the table. Nevertheless, this technique has some shortcomings. An intruder is still able to append a forged pattern to the verification table or replace someone encrypted password.

Li et al. (2001) proposed a remote password authentication scheme based on a *Neural Network* (*NN*), *Multi Layer Perceptron* (*MLP*) (Bishop 1995; Haykin 1998). This system identifies the legitimate user in real time and it is also applicable to a multiserver network architecture, where the input pattern is the user password and the output is the serviceable server. The authors compared the following supervised models: *Back-Propagation NN*, *Sum-of-Product Network, Hybrid Sum-of-Product Network*. Successively, Wang and Wang (2008) proposed to use a recurrent associative memory, *Hopfield* NN (HNN)-based methodology (Haykin 1998), that has all the advantages of the approach in (Li et al. 2001) but eliminates the disadvantages of the NN method, e.g., long training time and recall approximation. The overall authentication scheme, that incorporates the HNN, can recall information for a legal user *ID* and *password* instantly and accurately. The scheme can be used for any access control of computing

resources, such as multiple server access permission or role-based security control. Reyhani and Mahdavi (2007) trained a *Radial Basis Function* (*RBF*) NN (Haykin 1998) to store encrypted passwords. One of the advantages of RBF NNs is the faster training compared to the MLP network ones. However, it is worth remarking that in both works by Reyhani and Mahdavi (2007) and Wang and Wang (2008) during the authorization process the system must compare the NN-generated password with the one provided by the user determining the match. In this way, the password can be decrypted if the key is compromised or the encrypted password can be saved on an external storage. For this reason, Sibai et al. (2008, 2009) introduced a methodology that allows to perform the comparison by using a MLP NN without using a comparator module. In this case, the output of the trained NN is 1 for a (*user ID, password*) access authorization, or 0 for access denial. Finally, Singh (2009) gave a different methodology to design the security system by using NN having the intrusion detection capability too. This is possible by the analysis of several logical parameters associated with the user activities.

2.2 Password strength

A password is a secret string of characters used for authentication. Password *strength* is the measurement of the effectiveness of a password in resisting to guessing and brute force attacks. The key to a strong password is the length and the complexity, i.e., the number of individual characters and the number of characters that could potentially be used in its creation, respectively. Most password strength checking tools rates the password as *very weak, weak, moderate* or *medium* or *good, strong* and *very strong*. Salem et al. (2008) presented a SC based approach to measure strength of passwords that may help to prevent *dictionary, brute-force* and *shoulder surfing* attacks simultaneously. The authors proposed to use *Fuzzy Sets* (Zadeh 1994) for each of the following three factors:

- length of the password phrase (supposed to be adequately long to avoid time crack attacks and human memory);
- dictionary based passwords;
- entropy of the password (entropy estimates the time to crack the password[4]).

In that work (see Fig. 1) each input has three *membership functions* (e.g., length has values in linguistic variables *Short*, *Medium* and *Long*) and the composition is obtained

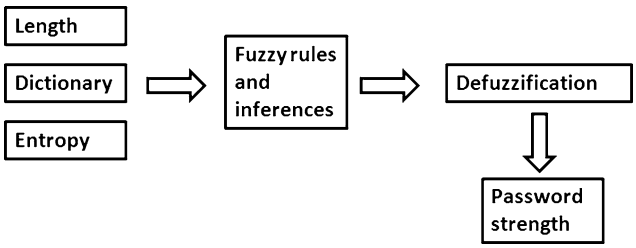

Fig. 1 Password strength obtained by using a fuzzy process with IF-THEN fuzzy rules as presented by Salem et al. (2008)

by *fuzzy inference systems*. The output is the password strength obtained after a *defuzzification* module (Zadeh 1994). Jamuna et al. (2009), in a further work, introduced a *Support Vector Machine* (*SVM*) (Shawe-Taylor and Cristianini 2004) based methodology to analyze the strength of the password. In particular, linear and nonlinear SVM classification models were considered and trained using the features extracted from a password dataset. Twenty-seven descriptive features were created as a fixed length vector. The performance of the model was evaluated using tenfold *cross validation*[5] (Hastie et al. 2001) and observed that SVM classifier trained with RBF kernel performs well (see also Suganya et al. 2010). Finally, Vijaya et al. (2009) compared several supervised ML techniques namely *C4.5 Decision tree classifier* (Hastie et al. 2001), MLP, *Naive Bayes Classifier* (Duda et al. 2000) and SVM to learn a model for password strength prediction. The results of the models were also compared with the existing password strength checking tools.

2.3 Proactive password checking

The password checking models are essentials for selecting hard to guess passwords. In particular, the passwords chosen by users are checked for possible weaknesses and, in this case, the user is asked to select a different password. Two different approaches exist for password checking:

- *Reactive* Programs such as "cracks" are run periodically by the system administrators to find weak passwords.
- *Proactive* When a user selects a password, the system checks immediately to verify whether it is acceptable.

[4] The entropy of a character set can be calculated by using Shannon estimation $H = -\sum P(x)\log_2 (P(x)^{-1})$ where $x \in$ *Character set*.

[5] In K-fold cross-validation the training set is partitioned into K subsets. Of the K subsets, a single subset is retained as the validation data for testing the model, and the remaining $K-1$ subsets are used as training data. The cross-validation process is then repeated K times, i.e. the folds, with each of the K subsets used exactly once as the validation data. The K results from the folds then can be averaged to produce a single estimation.

In password checking a disadvantage is the space required by dictionaries and the time required for checking. An interesting approach for designing a proactive password checker is the one proposed by Bergadano et al. (1998). They viewed the problem of password classification as a ML problem. The system, in a training phase, get the knowledge for distinguishing weak passwords from strong ones, by using dictionaries of examples for both weak and strong passwords. This knowledge was represented by means of a decision tree. The same technique was subsequently applied by Blundo et al. (2004). Therein the previous checker is improved by using a ML technique for the construction of the decision tree, i.e., the *Minimum Description Length* (MDL) (Hastie et al. 2001) principle. Substantially, the proactive password checker prevented the choice of easy-to-guess passwords using a decision tree constructed by applying the minimum description length principle and a *pessimistic pruning* technique. Moreover, Ruffo and Bergadano (2005) designed a new proactive password checking system (EnFilter). It is composed of a set of configurable filters that use *decision trees*, *lexical analysers*, as well as *Levenshtein distance* (Cormen et al. 2009) based techniques. Blundo et al. (2002) put forward the possibility of using NNs for proactive password checking. Instead of using standard computing techniques the classifier was implemented by means of a single *perceptron* (Haykin 1998). Successively, Ciaramella et al. (2006a) proposed to use MLP networks. The authors evaluated the performance of several network topologies and different pre-processing techniques e.g., *Principal Component Analysis* (*PCA*) (Bishop 1995) and compared the results with RBF NN and *Fuzzy Relational* (*FR*) NN (Ciaramella et al. 2006b) approaches. The solution has the main advantage that such checkers might be easily implemented using the smart card technology, too.

2.4 Password keystroke dynamics

In typing a phrase or a string of characters, the typing dynamics or timing pattern can be measured and used for identity verification. More specifically, a timing vector consists of the keystroke duration times interleaved with the keystroke interval times at the accuracy of milliseconds. Combining the keystroke dynamics identity with the password while users access computer systems is a considerably effective way to verify the valid access due to the unique keystroke dynamics of each person. Several works have been made by using MLP for keystroke dynamics identity. Lin (1997) and Cho et al. (2000) introduced MLP NNs to discriminate valid users and impostors according to each individual password keystroke pattern. However, the approach had some limitations:

1. it took too long to train the model;
2. data were preprocessed subjectively by a human;
3. a large data set was required.

Yu and Cho (2004) introduced a combination of approaches and models that could solve or alleviate these limitations. First of all, a SVM was proposed for *novelty detection*[6] in order to build a model of the owner keystroke dynamics and use this to detect imposters using similarity measures. A wrapper feature selection approach was employed which was able automatically to select a relevant subset of features and ignore the rest, thus producing a better accuracy. In particular, a *Genetic Algorithm* (*GA*) (Mitchell 1996) based wrapper approach is used. Finally, an *ensemble model* (Hastie et al. 2001) based on feature selection (FS-Ensemble) was proposed to alleviate the deficiency of a small training data set.

Other ML approaches, based on SVM have been used to address the classification problem presented by keystroke dynamics. Sang et al. (2005), de Oliveira et al. (2005) and Sung and Cho (2006) have applied SVM to a small keystroke dataset and compared their results to standard NN technology. Furthermore, ML techniques were presented by Kang et al. (2007) where the *k-means clustering* algorithm (Duda et al. 2000) was used whereas Revett et al. (2007) provided evidence that a *Probabilistic Neural Network* (PNN) (Haykin 1998) outperforms MLP in terms of reduced training time and classification accuracy. Zhao (2006) compared different ML classification methods as decision tree, *Naive Bayesian, Instance Based Learning, Decision Table, One Rule, Random Tree* and *K-star* (Duda et al. 2000). Compared with the conventional *Nearest Neighbour* method (Bishop 1995) these learning methods, especially decision trees, can be more accurate. Instead, the main objective of Killourhy and Maxion (2009) was to collect a keystroke-dynamics dataset, in order to develop a repeatable evaluation procedure, and to measure the performance of a range of 14 detectors from the keystroke dynamics and pattern-recognition literature. Finally, various *Fuzzy Logic* (Zadeh 1994) based algorithms have been proposed, too. For instance, Hussien et al. (1989) and de Ru and Eloff (1997) used a combination of *fuzzy clustering* algorithms (Lin and Lee 1996) depending on the number of acquired (*user ID, password*) samples. Haider et al. (2000) assigned ranges of typing times to fuzzy sets. Revett et al. (2005) used the *Rough Sets* (Pawlak 1982) induction algorithm to extract rules that form models for predicting the validity of a (*user ID, password*).

[6] Novelty detection is the identification of new or unknown data that a machine learning based system is not aware of during its training (Markov and Singh 2003).

3 Intrusion detection systems

As users, people depend on the Internet in their daily life for simple tasks such as checking e-mails, but also for managing private and financial information. However, sending such information through Internet also means that the network has become more and more an appealing place for hackers. To face this threat, the research community has answered with an increasing interest in Intrusion Detection, that results in developing new methods to timely detect intruders and prevent damages. Therefore *Intrusion Detection Systems* (*IDS*s) have become an indispensable component of security infrastructure to detect these threats before they inflict unrecoverable damages.

3.1 General overview of an IDS

Following Kruegel et al. (2004), "Intrusion detection is the process of identifying and responding to malicious activities targeted at computing and network resources". An IDS dynamically monitors the events taking place in a system, and decides whether these events are symptomatic of a violation or constitute a legitimate use of the system (Wu and Banzhaf 2010). Several kinds of violations are possible, ranging from the abuse of privileges to the use of attacks for exploiting software or protocol vulnerability. Intrusion detection can be considered as a ML problem, discriminating between network attacks and normal network behaviors or furtherly distinguishing between different categories of attacks. The typical organization of an IDS is shown in Fig. 2. Intrusion detection can be taxonomized into two families according to the kind of input information they analyze:

- *host-based detections* (HIDS), analyze host-bound audit sources such as operating system audit trails, system or application logs collected from the target host machine. Since the information provided by the audit data can be extremely rich and complex, host-based approaches can obtain high detection and low false-alarm rates. Nonetheless, disadvantages exist for host-based detections. Firstly, it is difficult for HIDSs to prevent attacks, since when an intrusion is detected, the attack is partially occurred. Secondly, audit data may be altered by attackers, influencing their reliability.
- *network-based detections* (NIDS), analyze network packets that travel through switches and routers. Although such information is not so rich as the audit data of the target host machine, there are advantages for network-based approaches, i.e., they can detect "distributed" intrusions over the whole network and thus lighten the burden on each individual host machine for detecting intrusions. Besides, NIDSs can defend the machine against attacks, as detection occurs before the data arrive at the machine.

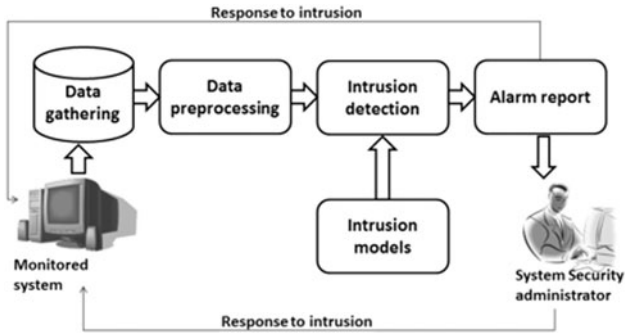

Fig. 2 Typical organization of an IDS: *thick arrows* indicate data and/or control flow, while *thin arrows* indicate responses to intrusive activities

Furthermore, intrusion detection can be divided in two broad categories (Hu et al. 2008), according to the detection methods they employ, namely:

- *misuse detection*, where a search for the traces or patterns of well-known attacks is carried out. Misuse detection has high detection rates for the well-known intrusions but fails to detect novel intrusions. An alarm is generated if a previously specified pattern is recognized. The strength of a misuse-based IDS lies in being highly accurate, i.e., it rarely raises an alarm due to a normal activity. On the other hand, its effectiveness depends on the completeness of the signatures. Therefore, a misuse-based system cannot recognize new attacks.
- *anomaly detection*, which involves the use of a model of a normal user or system behavior, usually known as *user* or *system profile*, and highlights significant deviations from this model as potentially malicious (Patcha and Park 2007). The main advantage results in detecting potentially attacks that have never been seen in advance (Owezarski et al. 2010). However, there exist cases in which events that deviate from the system profile are not necessarily malicious. The goodness of an anomaly detector is its ability to detect previously unknown attacks.

It is worth observing that the IDS classes discussed so far, represent a possible categorization. In fact many flavors of IDSs have been proposed leading to several IDS taxonomies (Debar et al. 1999). Thus, IDSs can be also classified according to the type of analysis ("real-time" or "offline"), the type of data processing ("centralized" or "distributed") or the response to the intrusion ("passive" or "active"). An

IDS aims to discriminate between intrusion attempts and normal activities. In doing so, however, an IDS can introduce classification mistakes. A *false positive* is a normal input for which the system erroneously raises an alert. A *false negative*, on the other hand, is a malicious input that the IDS fails to report. The correctly classified input data are usually referred to as *true positives* (attacks) and *true negatives* (normal traffic).

3.2 ML and SC approaches to IDS modeling

ML and SC techniques have been applied to IDSs at different levels (from network-based to host-based IDSs, from misuse to anomaly detection) (Tsai et al. 2009; Wu and Banzhaf 2010). It can divide the different ML and SC techniques in four families: supervised learning-based approaches, unsupervised learning-based approaches, statistical modeling-based approaches and ensemble-based approaches.

3.2.1 Supervised learning-based approaches

In this family NN and SVM based approaches have been mainly proposed.

– *NN-based approaches*: in early developments of IDSs (Wu and Banzhaf 2010), MLP NN were applied primarily to anomaly detection on user behavior level (Ryan et al. 1998; Tan 1995). Tan (1995) used command sets, CPU usage, login host addresses to discriminate between normal and abnormal behaviors, whereas Ryan et al. (1998) considered the pattern of commands and their frequency. In Ghosh et al. (1998) and Ghosh and Schwartzbard (1999) instead, a MLP was built to deal with software behavior described by sequences of system calls. A *leaky bucket* algorithm (Kurose and Ross 2010) was used to remember anomalous events diagnosed by the network, so that the temporal characteristics of program patterns were accurately captured. Research, such as Hofmann et al. (2003) and Rapaka et al. (2003), employed RBFs to learn multiple local clusters for well-known attacks and for normal events. This is because RBF NN are more suitable for problems with large sample size (Chan et al. 2005; Wu and Banzhaf 2010). *Recurrent* (*RNN*s) (i.e., *Elman model*, Haykin 1998) were used for detecting attacks spread over a period of time (Al-Subaie and Zulkernine 2007; Cheng et al. 2005). Chan et al. (2005) used a RNN to detect network anomalies exploiting the temporal locality property of the KDD99 network traffic dataset; whereas Al-Subaie and Zulkernine (2007) used a RNN classifier for the UNM system call dataset (Wu and Banzhaf 2010).

Han and Cho (2006) employed evolutionary NNs to detect intrusions. Zhang et al. (2005) proposed a hierarchical RBF NNs for intrusion detection. The same problem was tackled by Toosi and Kahani (2007) using the *Adaptive Neuro-Fuzzy Inference System* (*ANFIS*) (Lin and Lee 1996), as neuro-fuzzy classifier. The system was combined with a genetic algorithm to optimize the fuzzy decision-making engine obtained by the fuzzy inference approach.

– *SVM-based approaches* Mukkamala et al. (2002) and Sung and Mukkamala (2003) used SVMs for discriminating between normal network behaviors and intrusions and furtherly identify significative features for intrusion detection. Besides, Mill and Inoue (2004) proposed the TreeSVM and ArraySVM for solving the problem of inefficiency of the *sequential minimal optimization algorithm* (Platt 1999) for very large training dataset in intrusion detection. To cope with this problem, Zhang and Shen (2004) proposed an approach for online training of SVMs for real-time intrusion detection based on an improved text categorization model. Finally, Ren et al. (2009) combined SVM and rough sets to use Internet protocol for intrusion detection.

3.2.2 Unsupervised learning-based approaches

Supervised learning methods for intrusion detection can only detect known intrusions, namely only the intrusions that the system has seen and learnt previously when it has been undergone to the training. Unsupervised learning methods can detect the intrusions that have not been previously learned. Wang et al. (2006) and Hoglund et al. (2000) employed *Self-Organizing Maps* (*SOM*s) (Haykin 1998) to learn patterns of normal system activities from audit data. The previous approaches were enhanced by Sarasamma et al. (2005), who proposed a hierarchical SOM. In this way, they used the SOM classification capability on properly selected dimensions of the data set to detect anomalies. An alternative approach was suggested by Jiang et al. (2006) who employed an extension of the*K-means* (Duda et al. 2000) algorithm to detect intrusions. Siripanwattana and Srinoy (2008) combined rough sets and *fuzzy c-means* (Bezdek 1981) for anomaly-based network detection, whereas Srinoy et al. (2005) experimented an *Independent Component Analysis* (*ICA*) (Hyvärinen et al. 2001) for feature selection combined with a rough-fuzzy clustering. Both the approaches allow to recognize known attacks and to detect suspicious activity possibly resulting in a new, previously unknown, attack. In order to overcome the shortages of a single-level structure, which is only able to detect either misuse or anomaly attacks, Liu et al. (2007) designed a hierarchical intrusion detection model using

PCA neural network. Finally, Shon and Moon (2007), suggested a new SVM approach, named *Enhanced SVM*, which combines *soft-margin SVM* and *one-class SVM* (Shawe-Taylor and Cristianini 2004) in order to provide unsupervised learning and low false alarm capability to detect anomalies, similar to that of a supervised SVM approach.

3.2.3 Statistical modelling-based approaches

Statistical methods are the first techniques applied to IDS research, among ML and SC methods. According to this approach, the normal user behavior is described in terms on what is acceptable within the system usage policies. Using various statistical modeling techniques, user behavior is monitored and, if there is any deviation from predefined normal behavior, anomaly activity of users will be considered an attack. Farid and Rahman (2008) employed an adaptive *Bayesian algorithm* (Duda et al. 2000) to recognize different attack types on KDD99 data set. Alternatively, Qiao et al. (2002) applied a *Hidden Markov Model* (*HMM*) (Rabiner 1989) on system calls to detect anomalous intrusions. They identified various state transitions that a special UNIX based process goes through from the start to the end. The HMM was applied to all the collected system calls specific to that process. Using these state transition sequences, they built a database of normal sequences and then monitored system call sequences against the database to detect anomaly. Wright et al. (2004) proposed a formalization of the traffic exchange in terms of HMM *profiles* to classify traffic sequences at application level. Dainotti et al. (2008), instead, described a HMM-based packet-level model of traffic sources. In addition, a second fruitful application of the model (Dainotti et al. 2008) is the short-term prediction of future traffic behavior. *Behavioral models* for host-based intrusion detection have been proposed by Gao et al. (2006). The authors profiled the normal sequence of system calls and raised alarms whenever a sequence is unlikely to be seen. Finally, Sperotto et al. (2009) applied HMM to represent a flow time series model of *SSH brute-force attack* for successful emulating the attacker behavior.

3.2.4 Ensemble-based approaches

As previously discussed, many classification approaches for ML and SC have been applied to improve detection accuracy, and to reduce false positive errors, as well. Nevertheless, every approach has its strengths and weaknesses, resulting in various accuracy levels on different classes. Even models built by the same algorithm show differences in misclassification. Therefore, several authors (Abraham and Jain 2005; Abraham et al. 2007; Peddabachigari et al. 2007) investigated the possibility of assembling different learning approaches to detect intrusions. In these studies, they trained and tested a decision tree model, a *linear genetic program model* (Mitchell 1996), and a fuzzy classifier model on the KDD99 dataset, respectively. They observed in the experiments that different models provided complementary information about the patterns to be classified. Therefore, instead of using a single model to classify all classes, they selected the best model for each class, and then combined them in order to maximize computational efficiency and detection accuracy. Techniques, such as *majority vote* or *winner-takes-all* (Duda et al. 2000), are often used to decide the output of an *ensemble model* (Duda et al. 2000) when the predictions of different models conflict. Gudadhe et al. (2010) studied ensemble of decision trees, combined through *boosting* (Schapire 1990) to recognize attacks from network data flow. A special type of decision trees, namely *decision stumps*,[7] were combined by Hu et al. (2008). Each decision stump was used as weak classifier in order to obtain a strong classifier by means of *AdaBoost* (Freund and Schapire 1996). In addition, Panda and Patra (2009a, b) compared Adaboost with *Random Forest* (Hastie et al. 2001) and several rule based classifiers, in constructing efficient network intrusion detection models. Finally, Chan et al. (2005) used RBF NN to merge results from multiple classifiers.

4 Spam filtering

Unsolicited e-mail messages are called *spam* (Hoanca 2006; Castiglione et al. 2011). Spam mines the e-mail reliability and causes so large economical impact that some countries made a legislation to ban spamming (Talbot 2008). However, the legislation is often ineffective due the difficulties in identify the real sender of spam messages. A different approach against spamming consists in the use of *spam filters* (Goodman et al. 2007). After having identified a spam message, the spam filter can usually apply two different strategies. The former consists in moving the spam message in an appropriate folder containing only spam. The latter labels the message as spam leaving the user the decision of erasing the message. Early spam filters employed user-defined rules, identified on the basis of the knowledge of the regularities observed in spam message. Nevertheless, *spammers*, i.e., people that send spam e-mails, generally use tools (Stern 2008) that have the aim of minimizing the number of spam messages that may be identified by spam filters. For instance, spammers usually employ *obfuscation* techniques consisting in masking the

[7] Decision trees with a root node and two leaf nodes.

terms that are very frequent in the spam e-mails, e.g., replacing "free" with "fr€€". In this way, obfuscation techniques make harder the identification of a spam message by spam filters. At the present time, the former rule-based spam filters are replaced by ML and SC based filter tools. ML and SC techniques can extract knowledge from a limited collection of labeled messages where the labels are either "legitimate" or "spam". Then, ML and SC methods can use the extracted knowledge for classifying correctly new (i.e., not previously seen) e-mail messages. Given a collection of training labeled documents $\mathcal{T} \subseteq \mathcal{D}$, where $\mathcal{D}$ is the universe of all possible documents, the learning problem consists in estimating a decision function $f : \mathcal{D} \rightarrow \{0, 1\}$, where 0 corresponds to a legitimate message and 1 to a spam. ML and SC applications for spam filtering can be divided in two big families: text-based spam filtering and image-based spam filtering.

4.1 Text-based spam filtering

Text-based spam filtering can be viewed as a *text categorization* (TC) problem (Sebastiani 2002), i.e., assigning a text to a *class* (or category) previously defined. A typical text classification problem consists in assigning a news to a given category, e.g., sport, politics. Although TC is a good approach for handling text-based spam filtering, spam filtering has some proper characteristics. They should be taken into account if it wants that a ML and/or SC text-based spam filtering guarantees an adequate degree and fidelity to the real world. Main proper characteristics of spam filtering include (Fawcett 2003): skewed and changing class distributions, unequal and uncertain miss-classification costs of spam and legitimate messages, and the *concept drift*, namely a change in the significatives terms in spam messages.

Having said that, the usual structure of text-based spam filter is examined. The information contained in an e-mail message can be divided in two parts: the *header* and the *body* (Guzella and Caminhas 2009). The header contains the general information on the e-mail, i.e., the subject, the sender and the recipient; whereas the body contains the message. If it wants to use ML or/and SC methods in a text-based spam filter, some preprocessing steps have to be performed. The preprocessing steps can be divided in:

1. *tokenization*, that consists in extracting the words from the body of the message and eliminating the punctuation signs, i.e., fullstops, commas, parentheses, dashes and so on.
2. *stopping*, that remove *stop words*, that are the most frequent words in a message. Examples of stop words are articles, conjunctions, prepositions, auxiliary and modal verbs, adverbs.
3. *stemming* that reduces each word to its linguistic root, i.e., the *stem*. For instance, the stem of the word "going" is "go". Whereas, for the English language a reliable stemming algorithm (the *Porter's algorithm*, Porter 1980) is available, for other languages, e.g., the Italian language, a reliable stemming algorithm does not exist. In this case stemming process may be replaced with *lemmatization*, which consists in reducing the word to its *lemma*.[8]
4. *representation*, which converts the set of remaining words present in the message to a format, suitable to be processed by the selected ML and/or SC algorithm.

4.1.1 Representation of text

The representation of text in TC is crucial and therefore assumes the same relevance in text-based spam filtering. A very popular representation is the so-called *Bag of Words* (*BOW*) also known as the *vector-space model* (Salton et al. 1975).

Given a set of terms $\mathcal{W} = (w_1, w_2, \ldots, w_n)$ *a priori* chosen, a document d is represented as a N-dimensional feature vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, with $x_i = g(f(w_i)), i = 1, \ldots, N$, where $f(w_i)$ is the occurrence of the term w_i in the document d and $g(\cdot)$ is an appropriate function. Another possible approach consists in using n-gram word models (Zorkadis and Karras 2006), which considers sequences of words. For instance, if it considers the sequence "soft computing algorithms", the word 2-gram are "soft computing" and "computing algorithms". In a binary representation, a feature x_i is equal to 1 if the term w_i occurs in d, and 0 otherwise. A more popular feature representation is *tf-idf*, i.e., *term frequency-inverse document frequency* (Sebastiani 2002), where the feature x_i associated to the term w_i in the document (message) $d \in \mathcal{T}$ is given by:

$$x_i = n_{w_i d} \log\left(\frac{|\mathcal{T}|}{n_{w_i}}\right),$$

where $n_{w_{id}}$ is the number of occurrences of the term w_i in the document d and n_{w_i} the number of occurrences in the term w_i.

Given a set $\hat{\mathcal{W}}$ composed of all the terms from the collection of training documents $\mathcal{T}$, before the representation of the document, it is usual to apply some feature selection algorithm which aims to identify a subset $\mathcal{W} \subseteq \hat{\mathcal{W}}$ containing only the most representative terms. Several methodologies for TC have been proposed (Sebastiani 2002). All these methods compute a score for each term

[8] In linguistics the lemma of a word is the word that is conventionally chosen to represent all flexed forms of a given term. For instance, the lemma of the verb "am" is "to be".

and the terms are sorted on the score basis. Finally, the top N terms are picked, where N is a value *a priori* fixed. The *information gain* (or *mutual information*, Zhang et al. 2004) is the most popular feature selection algorithm. The information gain algorithm assigns to each term w_i the score $\tau(w_i)$ given by:

$$\tau(w_i) = \sum_{c\in\{0,1\}} \sum_{t\in\{w_i,w_i'\}} P(t,c) \log \left[\frac{P(t,c)}{P(t)P(c)}\right]$$

where w_i' denotes the absence of w_i. Finally, it is necessary to underline some points of weakness of the BOW approach, even if it is the most popular one. The most relevant weakness of the BOW approach is the concept drift that is a peculiar characteristic of spam. Since the structure of the feature vector is constant, i.e., has always the same elements, a term that was not initially included, during the model construction, cannot be considered in future (Gabrilovitch and Markovitch 2007). For instance, it supposes that it has trained a classifier with feature vectors containing three features related to the terms "bank", "euro", "account". This classifier cannot consider either variations of three terms in new e-mail messages, such as "€uro" or new terms such as "win". In order to consider a new term, it needs to retrain the classifier, that might be cumbersome if carried out frequently.

4.1.2 *Classifiers for spam filtering*

The first classifier proposed for spam filtering was the *naive Bayes classifier* (Duda et al. 2000). The Bayes classifier was introduced by Sahami et al. in the Technical Report WS-98-05. They tackled the problem of spam filtering in a Bayesian framework estimating the probability that a given message is spam. The Bayesian framework allows to integrate evidence from different sources. In this way, the authors also used non-textual features, e.g., the time when the e-mail was sent, in addition to the usual textual features. Later on, SVMs were applied by Drucker et al. (1999) to spam filtering. They used the BOW representation with binary or *tf-idf* features, selected by means of the information gain on private corpora. Compared with *boosting* based on *decision trees*, SVMs showed a slighty higher false positive rate but more robustness to different corpora and preprocessing procedures and required less time for the training. Moreover, SVMs required no feature selection procedure since can cope with a large number of features. The best results were obtained using a binary representation for SVMs and frequency-based for boosting. Clark et al. (2003) proposed LINGER, which is based on a MLP for e-mail categorization and spam filtering. Message were represented as BOW with feature selection based on information gain. Experiments, performed on public corpora, showed that LINGER outperformed the naive Bayes classifier obtaining very small false positive rates. Nevertheless, when the spam filter was trained and tested on different datasets the results degraded notably. Goodman and Yih (2006) proposed a *logistic regression model* (Hastie et al. 2001) for SF. They used binary features, taking into account if the feature occurs on the message header or body. The system developed, tested on two public corpora, obtained competitive results with the best spam filters. Oda and White (2003) proposed an *Artificial Immune System* (de Castro and Timmis 2002) for spam filtering. The system yielded promising results, namely true positive and negative rates of 90 and 99%, respectively. Carreras and Marquez (2001) used a variant of Adaboost, with decision trees as base classifiers and binary features. Adaboost tested on a public corpus outperforms decision trees and the naive Bayes classifier. Finally, Zhao and Zhang (2005) applied rough sets to classify messages into three classes: *spam*, *legitimate* and *suspicious*. In the first step, features are selected from the training set. Then a set of rules is deduced by means of a genetic algorithm. The set of rules divide the universe of messages into three regions. Tested on a public corpus outperformed the naive Bayes classifier.

4.2 Image-based spam filtering

Given the increasing number of spam messages containing images, spam filters need not only handle textual content but they also examine the image content. Aradhye et al. (2005) developed a system for identifying images characteristic of spam messages, based on the extraction of image regions containing text. It is faster than an OCR[9] since it needs not the text identification. They used a polynomial kernel SVM, fed by features, such as the percentage of the message containing text. This is selected for discriminating between images of spam messages and images (e.g., photos) included in legitimate messages. The results obtained on a private corpora indicated that the system can identify 70–80% of spam messages and 70–100% of legitimate spam containing images. Wu et al. (2005) proposed an approach for analyzing visual features of images attached to e-mails. The features extracted include, for example, the number of regions containing embedded text and the number of images containing such text. Due to the difficulties in collecting legitimate e-mails with attached images, the authors employed, as classifier, one-class SVM. Using a test composed of synthetic legitimate messages artificially generated and real spam e-mails with attached images, one-class SVM, compared with the usual two-class SVM, showed a

[9] OCR stands for Optical Character Recognizer.

Table 1 For each category of methods the number of works related to security password-based schemes for access control (SPBS), intrusion detection systems (IDs) and spam filtering (SF) are reported

ML and SC methods	SPBS	IDs	SF
Neural networks	16	17	1
Kernel methods	7	6	4
(Neuro) fuzzy and rough sets	5	5	1
Decision trees	5	2	1
Evolutionary systems	1	3	1
Bayesian classifiers, ICA/PCA, HMM	3	7	2
Ensemble methods	2	7	2
K-means, K-NN	2	1	0
Others	5	2	1

In the category Others are reported the methods not represented in the remaininig categories

reduced number of false positive despite of the increase of the number of false negatives. Finally, Fumera et al. (2006) proposed a system for carrying out a semantic analysis of the text embedded in the images attached to e-mails. They used an OCR for extracting the text and then performed the analysis of the content by means of a linear SVM, fed with features selected on the basis of the information gain. Experiments, driven on a public corpus (e.g., SpamAssassin corpus), showed that the system, using only the text extracted from the images, yielded the smallest false negative rate and low false positive rates ($\sim$1%). The authors concluded that considering the text embedded in the images can improve notably the performances of the spam filters.

5 Conclusions

This work overviewed ML and SC applications for three hot topics in ICT, i.e., security password-based schemes for access control, intrusion detection and spam filtering. The different number of ML and SC techniques applied to the previous problems are summarized in Table 1. It has been shown that ML and SC have been widely applied in ICT security, due to the fact that they allow a system to reply to changeable real-world inputs and learn to identify undesiderable behaviors. In this way, ML and SC are becoming more and more a very important tool for Computer Security. Unlike other ML and SC applications, the ones in the Computer Security have to work under *adversarial conditions* (Barreno et al. 2010). Adversarial conditions mean, for instance, that an attacker can attempt to use the adaptive aspect of ML and SC systems to cause them to fail. Developing *secure learning* algorithms, i.e., algorithms that can work under more disparate adversarial conditions is a big challenge for ML and SC researchers in the next future.

References

Abraham A, Jain R (2005) Soft computing models for network intrusion detection systems. In: Classification and clustering for knowledge discovery. Springer, Berlin, pp 191–207

Abraham A, Jain R, Thomas J, Han SY (2007) D-scids: distributed soft computing intrusion detection system. J Netw Comput Appl 30(1):81–89

Al-Subaie M, Zulkernine M (2007) The power of temporal pattern processing in anomaly intrusion detection. In: IEEE international conference on communications (ICC'07), pp 1391–1398

Aradhye H, Myers G, Herson J (2005) Image analysis for efficient categorization of image-based spam e-mail. In: Proceedings of the international conference on document analysis and recognition, pp 914–918

Barreno M, Nelson B, Joseph AD, Tygar JD (2010) The security of machine learning. Mach Learn 81:121–148

Bergadano F, Crispo B, Ruffo G (1998) High dictionary compression for proactive password checking. ACM Trans Inform Syst Secur 1(1):3–25

Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York

Bishop C (1995) Neural networks for pattern recognition. Cambridge University Press, Cambridge

Bishop C (2006) Pattern recognition and machine learning. Springer, Berlin

Blundo C, D'Arco P, De Santis A, Galdi C (2002) A novel approach to proactive password checking. In: Proceedings of INFRASEC 2002. LNCS, vol 2437. Springer, Berlin, pp 30–39

Blundo C, D'Arco P, De Santis A, Galdi C (2004) Hippocrates: a new proactive password checker. J Syst Softw 71(1–2):163–175

Carreras X, Marquez L (2001) Boosting trees for anti-spam email filtering. In: Proceedings of the 4th international conference on recent advances in natural language processing, pp 58–64

Castiglione A, De Santis A, Fiore U, Palmieri F (2011) An asynchronous covert channel using spam. Comput Math Appl (in press)

Chan APF, Ng WWY, Yeung DS, Tsang ECC (2005) Comparison of different fusion approaches for network intrusion detection using ensemble of rbfnn. In: Proceedings of international conference on machine learning and cybernetics, pp 3846–3851

Cheng E, Jin H, Han Z, Sun J (2005) Network-based anomaly detection using an elman network. In: Proceedings of INFRASEC 2002. LNCS, vol 3619. Springer, Berlin, pp 471–480

Cho S, Han C, Han DH, Kim H-I (2000) Web based keystroke dynamics identity verification using neural network. J Org Comput Electr Comm 10(4):295–307

Ciaramella A, D'Arco P, De Santis A, Galdi C, Tagliaferri R (2006a) Neural network techniques for proactive password checking. IEEE Trans Dependable Secur Comput 3(4):327–339

Ciaramella A, Tagliaferri R, Pedrycz W, Di Nola A (2006b) Fuzzy relational neural network. Int J Approx Reason 41(2):146–163

Clark J, Koprinska I, Poon J (2003) A neural network based approach to automated e-mail classification. In: Proceedings of the IEEE/WIC international conference on web intelligence

Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to algorithms. MIT Press, Cambridge

Dainotti A, Pescapé A, Rossi PS, Palmieri F, Ventre G (2008) Internet traffic modeling by means of hidden markov models. Comput Netw 52(14):2645–2662

de Castro LN, Timmis J (2002) Artificial immune systems: a new computational intelligence method. Springer, Berlin
de Oliveira M, Kinto VSE, Hernandez EDM, de Carvalho TC (2005) User authentication based on human typing patterns with artificial neural networks and support vector machines. In: Proceedings of SBC 2005
de Ru WG, Eloff J (1997) Enhanced password authentication through fuzzy logic. IEEE Expert 12(6):38–45
Debar H, Dacier M, Wespi A (1999) Towards a taxonomy of intrusion-detection systems. Comput Netw 31(9):805–822
Drucker H, Wu D, Vapnik VN (1999) Support vector machines for spam categorization. IEEE Trans Neural Netw 10(5):1048–1054
Duda RO, Hart PE, Stork DG (2000) Pattern classification. Wiley, New York
Farid DM, Rahman MZ (2008) Learning intrusion detection based on adaptive bayesian algorithm. In: Proceedings of International Conference on Computer and Information Technology (ICCIT 2008), pp 652–656
Fawcett T (2003) In: "vivo" spam filtering: a challenge problem for kdd. SIGKDD Explor 5(2):140–148
Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm. In: Proceedings of international conference in machine learning, pp 138–146
Fumera G, Pillai I, Roli F (2006) Spam filtering based on the analysis of text informations embedded into images. J Mach Learn Res 7:2699–2720
Gabrilovitch E, Markovitch S (2007) Harnessing the expertise of 70000 human editors: knowledge-based feature generation of text categorization. J Mach Learn Res 8:2297–2345
Gao D, Reiter MK, Song DX (2006) Behavioral distance measurement using hidden markov models. In: Proceedings of 9th international symposium recent advances in intrusion detection, pp 19–40
Ghosh AK, Schwartzbard A (1999) A study in using neural networks for anomaly and misuse detection. In: Proceedings of the 8th USENIX security symposium, pp 141–152
Ghosh AK, Wanken J, Charron F (1998) Detecting anomalous and unknown intrusions against programs. In: Proceedings of the 14th annual computer security applications conference (ACSAC'98), pp 259–267
Goodman J, Cormack GV, Heckerman D (2007) Spam and the ongoing battle for the inbox. Commun ACM 50(2):24–33
Goodman J, Yih W (2006) Online discriminative spam filter training. In: Proceedings of third conference on Email and anti spam
Gudadhe M, Prasad P, Wankhade K (2010) A new data mining network intrusion detection model. In: Proceedings of the international conference on computer and communication technology, pp 731–735
Guzella TS, Caminhas WM (2009) A review of machine learning approaches to spam filtering. Expert Syst Appl 36(7):10206–10222
Haider S, Abbas A, Zaidi AK (2000) A multi-technique approach for user identification through keystroke dynamics. In: Proceedings of the IEEE international conference on systems, man and cybernetics, pp 1336–1341
Han SJ, Cho SB (2006) Evolutionary neural networks for anomaly detection based on the behavior of a program. IEEE Trans Syst Man Cybern Part B Cybern 36(3):559–570
Hastie T, Tibshirani RJ, Friedman J (2001) The elements of statistical learning. Springer, Berlin
Haykin S (1998) Neural networks: a comprehensive foundation. Prentice Hall, Englewood Cliffs
Hoanca B (2006) How goods are our weapons in the spam wars? IEEE Technol Soc Mag 25(1):22–30
Hofmann A, Schmitz C, Sick B (2003) Rule extraction from neural networks for intrusion detection in computer networks. In: Proceedings of the IEEE international conference on systems, man and cybernetics, pp 1259–1265
Hoglund AJ, Hatonen K, Sorvari AS (2000) A computer host-based user anomaly detection system using the self-organizing map. In: Proceedings of the IEEE INNS-ENNS international joint conference on neural networks (IJCNN'00), pp 411–416
Hu W, Hu W, Maybank S (2008) Adaboost-based algorithm for network intrusion detection. IEEE Trans Syst Man Cybern Part B Cybern 38(2): 577–583
Hussien B, Bleha S, McLaren R (1989) An application of fuzzy algorithms in a computer access security system. Patt Recogn Lett 9:39–43
Hyvärinen A, Karhunen J, Oja E (2001) Independent component analysis. Wiley, New York
Jamuna KS, Karpagavalli S, Vijaya MS (2009) Novel approach for password strength analysis through support vector machine. Int J Recent Trends Eng 2(1):79–82
Jiang S, Song X, Wang H, Han J, Li Q (2006) A clustering-based method for unsupervised intrusion detection. Patt Recogn Lett 27(7):802–810
Kang P, Hwang S, Cho S (2007) Continual retraining of keystroke dynamics based authenticator. In: Proceedings of the 2nd International Conference on Biometrics (ICB' 07). Springer, Berlin, pp 1203–1211
Killourhy KS, Maxion RA (2009) Comparing anomaly-detection algorithms for keystroke dynamics. In: IEEE/IFIP International conference on dependable systems & networks, pp 125–134
Kruegel C, Valeur F, Vigna G (2004) Intrusion detection and challenges and solutions. Springer, Berlin
Kurose JF, Ross KW (2010) Computer networking. Addison Wesley, Reading
Li L-H, Lin I-C, Hwang M-S (2001) A remote password authentication scheme for multiserver architecture using neural networks. IEEE Trans Neural Netw 12(6):1498–1504
Lin DT (1997) Computer-access authentication with neural network based keystroke identity verification. In: Proceedings of international Conference on Neural Networks, pp 174–178
Lin C-T, Lee CSG (1996) Neural fuzzy systems. Prentice Hall, Englewood Cliffs
Liu G, Yi Z, Yang S (2007) A hierarchical intrusion detection model based on the pca neural networks. Neurocomputing 70:1561–1560
Markov M, Singh S (2003) Novelty detection: a review, part 1: statistical approaches. Signal Process 83:2481–2497
Mill J, Inoue A (2004) Support vector classifiers and network intrusion detection. In: Proceedings of international conference on fuzzy systems, pp 407–410
Mitchell M (1996) An introduction to genetic algorithms. MIT Press, Cambridge
Mukkamala S, Janoski G, Sung AH (2002) Intrusion detection using neural networks and support vector machines. In: Proceedings of the international joint conference on neural networks, pp 1702–1707
Oda T, White T (2003) Developing an immunity to spam. In: GECCO'03 Proceedings of the 2003 international conference on genetic and evolutionary computation, Part I. LNCS, vol 2723, Springer, pp 231–242
Owezarski P, Mazel J, Labit Y (2010) 0day anomaly detection made possible thanks to machine learning. In: Proceedings of the 8th international conference on WWIC 2010, pp 327–338
Panda M, Patra M (2009a) Ensembling rule based classifiers for detecting network intrusions. In: Proceedings of the international conference on advances in recent technologies in communication and computing, pp 19–22
Panda M, Patra M (2009b) Evaluating machine learning algorithms for detecting network intrusions. Int J Recent Trends Eng 1:472–477

Patcha A, Park J-M (2007) An overview of anomaly detection techniques: existing solutions and latest technological trends. Comput Netw 51:3448–3470

Pawlak Z (1982) Rough sets. Int J Parall Program 11(5):341–356

Peddabachigari S, Abraham A, Grosan C, Thomas J (2007) Modeling intrusion detection system using hybrid intelligent systems. J Netw Comput Appl 30(1):114–132

Platt JC (1999) Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel methods, pp 185–208. MIT Press, Cambridge

Porter MF (1980) An algorithm for suffix stripping. Program 14(3):77–84

Qiao Y, Xin XW, Bin Y, Ge S (2002) Anomaly intrusion detection method based on hmm. IEEE Electr Lett 38(13):663–664

Rabiner L (1989) A tutorial on hidden markov models and selected applications in speech recognition. In: Readings in speech recognition, pp 267–299

Rapaka A, Novokhodko A, Wunsch D (2003) Intrusion detection using radial basis function network on sequence of system calls. In: Proceedings of the international joint conference on neural networks, pp 1820–1825

Ren X, Wang R, Zhou H (2009) Intrusion detection method using protocol classification and rough set based support vector machine. Comput Inform Sci 2(4):100–108

Revett K, Gorunescu F, Gorunescu M, Ene M, de Magalhaes ST, Santos HMD (2007) A machine learning approach to keystroke dynamics based user authentication. Int J Electr Secur Dig Forens 1(1):55–70

Revett K, Magalhaes S, Santos H (2005) Developing a keystroke dynamics based agent using rough sets. In: The 2005 IEEE/WIC/ACM international joint conference on web intelligence and intelligent agent technology Compiegne, pp 56–61

Reyhani SZ, Mahdavi M (2007) User authentication using neural network in smart home networks. Int J Smart Home 1(2):147–154

Ruffo G, Bergadano F (2005) Enfilter: a password enforcement and filter tool based on pattern recognition techniques. In: 13th international conference of image processing (ICIAP 2005). LNCS, vol 3617, Springer, Berlin, pp 75–82

Ryan J, Lin MJ, Miikkulainen R (1998) Intrusion detection with neural networks. In: Advances in neural information processing systems, vol 10. MIT Press, Cambridge, pp 943–949

Salem O, Hossain A, Kamala M (2008) Intelligent system to measure the strength of authentication. In: Proceedings of 3rd international conference on information and communication technologies: from theory to applications, pp 1–6

Salton G, Wong A, Yang CS (1975) A vector-space model for automatic indexing. Commun ACM 18(11):613–620

Sang Y, Shen H, Fan P (2005) Novel impostors detection in keystroke dynamics using support vector machines. In: Parallel and distributed computing: applications and technologies, pp 666–669. LNCS, vol 3320. Springer, Berlin

Sarasamma ST, Zhu QA, Huff J (2005) Hierarchical kohonen net for anomaly detection in network security. IEEE Trans Syst Man Cybern Part B Cybern 35(2):302–312

Schapire RE (1990) The strength of weak learnability. Mach Learn 5(2):197–227

Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47

Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge

Shon T, Moon J (2007) A hybrid machine learning approach to network anomaly detection. Inform Sci 177:3799–3821

Sibai FN, Shehhi A, Shehhi S, Shehhi B, Salami N (2008) Secure password detection with artificial neural networks. In: Proceedings of the international conference on innovations in information technology, pp 628–632

Sibai FN, Shehhi A, Shehhi S, Shehhi B, Salami N (2009) Designing and training feed-forward artificial neural networks for secure access authorization. In: Pattern recognition. InTech, Rijeka, pp 666–669

Singh MK (2009) Password based a generalize robust security system design using neural network. Int J Comput Sci Iss 4(2):1–9

Siripanwattana W, Srinoy S (2008) Information security based soft computing techniques. In: Proceedings of international multi-Conference of engineers and computer scientists

Sperotto A, Sadre R, de Boer P-T, Pras A (2009) Hidden markov model modeling of ssh brute-force attacks. In: Proceedings of the 20th IFIP/IEEE international workshop on distributed systems: operations and management: integrated management of systems, services, processes and people in IT, pp 164–176

Srinoy S, Kurutach W, Chimphlee W, Chimphlee S (2005) Network anomaly detection using soft computing. World Acad Sci Eng Technol 9:140–144

Stahl B, Elizondo D, Carroll-Mayer M, Zheng Y, Wakunuma K (2010) Ethical and legal issues of the use of computational intelligence techniques in computer security and computer forensics. In: Proceedings of The 2010 international joint conference on neural networks (IJCNN), pp 1–8

Stern H (2008) A survey of modern tools. In: Proceedings of the fifth conference on email and anti-spam

Suganya G, Karpgavalli S, Christina V (2010) Proactive password strength analyzer using filters and machine learning techniques. Int J Comput Appl 7(14):1–5

Sung KS, Cho S (2006) Ga svm wrapper ensemble for keystroke dynamics authentication. In: International conference on Biometrics, pp 654–660

Sung AH, Mukkamala S (2003) Identifying important features for intrusion detection using support vector machines and neural networks. In: Proceedings of the 2003 symposium on applications and the internet, pp 209–216

Talbot D (2008) Where spam is born. Technol Rev 111(3):28–28

Tan K (1995) The application of neural networks to unix computer security. In: Proceedings of IEEE international Conference on Neural Networks, pp 476–481

Toosi AN, Kahani M (2007) A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. Comput Commun 30:2201–2212

Tsai C, Hsu Y, Lin C, Lin W (2009) Intrusion detection by machine learning: a review. Expert Syst Appl 36:11994–12000

Vijaya MS, Jamuna KS, Karpagavalli S (2009) Password strength prediction using supervised machine learning techniques. In: 2009 international conference on advances in computing, control, and telecommunication technologies, pp 401–405

Wang S, Wang H (2008) Password authentication using hopfield neural networks. IEEE Trans Syst Man Cybern Part C Appl Rev 38(2):265–268

Wang W, Guan X, Zhang X, Yang L (2006) Profiling program behavior for anomaly intrusion detection based on the transition and frequency property of computer audit data. Comput Secur 25(7):539–550

Weiser M (1991) The computer for the twenty-first century. Scientific American, New York, pp 94–100

Wright CV, Monrose F, Masson GM (2004) Hmm profiles for network traffic classification. In: Proceedings of the 2004 ACM workshop on visualization and data mining for computer security (VizSEC/DMSEC'04), pp 9–15

Wu SX, Banzhaf W (2010) The use of computational intelligence in intrusion detection systems: a review. Appl Soft Comput 10:1–35

Wu C-T, Cheng K-T, Zhu Q, Wu Y-L (2005) Using visual features for anti-spam filtering. In: Proceedings of the IEEE international conference on image processing, pp 509–512

Yu E, Cho S (2004) Keystroke dynamics identity verification problems and practical solutions. Comput Secur 23(5):428–440
Zadeh LH (1994) Fuzzy logic, neural networks and soft computing. Commun ACM 37(3):77–84
Zhang C, Jiang J, Kamel M (2005) Intrusion detection using hierarchical neural networks. Patt Recogn Lett 26(6):779–791
Zhang L, Zhu J, Yao T (2004) An evaluation of statistical spam filtering techniques. ACM Trans Asian Lang Inform Process 3(4):243–269
Zhang Z, Shen H (2004) Online training of svms for real-time intrusion detection. In: Proceedings of the 18th international conference on advanced information networking and applications, pp 568–573
Zhao W, Zhang Z (2005) An email classification model based on rough set theory. In: Proceedings of the international conference on active media technology
Zhao Y (2006) Learning user keystroke patterns for authentication. World Academy of Science Engineering and Technology, vol 14
Zorkadis V, Karras DA (2006) Efficient information theoretic extraction of higher order feature for improving neural network-based spam e-mail categorization. J Exp Theor Artif Intell 18(4):523–534