Nature-Inspired Techniques in the Context of Fraud Detection

Article *in* IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews) · November 2012

DOI: 10.1109/TSMCC.2012.2215851

CITATIONS
66

READS
1,155

4 authors, including:

Mohammad Behdad
University of Western Australia
8 PUBLICATIONS **99** CITATIONS

SEE PROFILE

Mohammed Bennamoun
University of Western Australia
591 PUBLICATIONS **16,152** CITATIONS

SEE PROFILE

Tim French
University of Western Australia
91 PUBLICATIONS **755** CITATIONS

SEE PROFILE

Nature-Inspired Techniques in the Context of Fraud Detection

Mohammad Behdad, Luigi Barone, Mohammed Bennamoun, and Tim French

***Abstract*—Electronic fraud is highly lucrative, with estimates suggesting these crimes to be worth millions of dollars annually. Because of its complex nature, electronic fraud detection is typically impractical to solve without automation. However, the creation of automated systems to detect fraud is very difficult as adversaries readily adapt and change their fraudulent activities which are often lost in the magnitude of legitimate transactions. This study reviews the most popular types of electronic fraud and the existing nature-inspired detection methods that are used for them. The common characteristics of electronic fraud are examined in detail along with the difficulties and challenges that these present to computational intelligence systems. Finally, open questions and opportunities for further work, including a discussion of emerging types of electronic fraud, are presented to provide a context for ongoing research.**

***Index Terms*—Evolutionary computation, fraud, pattern analysis, security.**

I. Introduction

FRAUD is the general term which is used to describe actions that are surreptitiously undertaken by one party to obtain an unjust advantage over another. Typically, the perpetrator, using false or misleading representations, attempts to disguise their activities in order to avoid detection for as long as possible, in an effort to maximize the effects of their fraudulent behavior.

This research paper deals with electronic fraud. As the name suggests, electronic fraud (or e-fraud for short) is a fraudulent action that is perpetrated using electronic technologies and devices, such as computers, Internet, email, telephone networks, and mobile phones.

Nowadays, most organizations are doing increasingly large portions of their tasks electronically, for example, accounting, human resources, customer relationship management, marketing, and sales. As computers have become cheaper and omnipresent, and mobile phones have become as powerful as a simple PC, individuals have also started using computers and mobile phones to do most of their banking, shopping, and entertainment. This increasing trend in computerization has lured criminals to move their efforts into this new territory. This move has turned out to be very lucrative with the rate to adopt new technology much higher than the rate to develop new security and defense mechanisms by computer experts.

TABLE I
IC3 Report on Internet Crime. Source: [2]

Year	Complaints Received	Dollar Loss
2009	336,655	$559.7 million
2008	275,284	$265 million
2007	206,884	$239.09 million
2006	207,492	$198.44 million
2005	231,493	$183.12 million

The cost of fraud for individuals and companies is quite significant. In 2008, the Australian Bureau of Statistics released a report which states that the cost of personal fraud for Australians in 2007 was $980 million [1]. As another example, the Internet Crime Complaint Center (IC3), which is a joint operation between the Federal Bureau of Investigation (FBI) and the National White Collar Crime Center (NW3C), in its 2009 Internet Crime Report states "online crime complaints increased substantially once again last year, a 22.3% increase from 2008. The total loss linked to online fraud was $559.7 million; this is up from $265 million in 2008" [2]. Table I summarizes the number of complaints that are received by the IC3 between 2005 and 2009 and the corresponding dollar losses. Further, as David Kirk, Director of Fraud Prosecution Services, states "as night follows day, fraud increases in a recession" [3], and hence with the recent downturn in the global economic market, the problem of fraud is likely to be on the increase.

The huge monetary loss endured by companies and individuals who are victims of fraud have fueled research in developing defense mechanisms against it. The first step to secure systems in this regard is preventing fraud. Put simply, fraud prevention is the task to stop fraud from happening in the first place. It is done by improving technologies and designs. However, this approach is not always successful and is occasionally infiltrated by fraudsters. The next layer of defense is fraud detection. Fraud detection tries to detect and recognize fraudulent activities as they enter systems and report them to a system manager.

Different branches and techniques of artificial intelligence and statistics have been and are being used in e-fraud detection systems, among which is nature-inspired techniques. Nature-inspired techniques, as the name implies, are artificial intelligence techniques which are inspired by the way natural systems work. For instance, the ant colony optimization (ACO) is based on the way ants find the shortest path between their nest and the food source [4], or evolutionary algorithms (EAs) are inspired by natural evolution [5]. Nature-inspired techniques are very effective in solving classification and optimization problems. The first reason for this success is their versatility [6].

Manuscript received September 13, 2011; revised May 28, 2012; accepted August 05, 2012. Date of current version December 17, 2012. The work of M. Behdad was supported by the Robert and Maude Gledden Postgraduate Scholarship. This paper was recommended by Associate Editor N. Wu.

The authors are with the Department of Computer Science and Software Engineering, The University of Western Australia, Crawley W.A. 6009, Perth, Australia (e-mail: behdad@csse.uwa.edu.au; luigi@csse.uwa.edu.au; bennamou@csse.uwa.edu.au; tim@csse.uwa.edu.au).

Digital Object Identifier 10.1109/TSMCC.2012.2215851

That is, they can be used for any problem with less dependence on the domain knowledge. They make few or no assumptions about the problem and can search very large spaces of candidate solutions. The second reason for their success is their robustness [7]. They can solve optimization problems in the presence of a wide range of uncertainties such as noise in the environment [8]. Adaptability is another valuable characteristic of nature-inspired techniques [7]. Many artificial intelligence methods, which have a centralized control, have difficulty in dealing with changing and dynamic environments. However, nature-inspired techniques maintain a population of diverse individuals, similar to natural systems or ecosystems, each of which is good at something, and are flexible and capable of changing in response to changes in their environment. The last important characteristic of nature-inspired techniques to mention is their capability in specialization. They use different parts of the population for different tasks. In this manner, these techniques have specialized soldiers for each group of tasks. As an example, XCS, which is a version of learning classifier systems (LCSs), uses a niche-based method in maintaining its population of rules [9]. All these characteristics make nature-inspired techniques very powerful methods to use in an e-fraud detection system.

This paper surveys the nature-inspired approaches for automated fraud detection and the challenges fraud presents to automated methods. In Section II, the common properties of fraudulent activities are introduced. Section III discusses the most important types of electronic fraud. Then, in Section IV, the most commonly used nature-inspired techniques for e-fraud detection are examined. Section V reviews the techniques that are available to improve the performance of nature-based techniques to deal with different challenges of fraud. Finally, Section VI concludes this study by reviewing some of the emerging types of e-fraud.

II. Properties of Fraud

Fraud detection is an extremely difficult task. Since off-the-shelf solutions are rarely found to detect fraud, organizations are finding it hard to address this problem and hence are incurring heavy losses [1]. The reason for this failure is the fact that fraud detection must deal with some uniquely challenging properties, which are described later. In Section III, we specifically focus on the most prevalent types of e-fraud, such as email spam and network intrusion.

A. Experience Imbalance

E-fraud detection involves learning a classifier from a labeled training set. This training set contains instances which are labeled positive (fraud) and negative (nonfraud). Perhaps, the most defining and important characteristic of fraudulent activity is the imbalance in positive and negative experiences that are presented during learning; i.e., the proportion of fraudulent records to nonfraudulent records is vastly skewed. This "needle-in-a-haystack" like effect typically leads to unacceptably high false-positive rates in the data [10]. Indeed, these "rare" instances can have significant negative impacts on performance in data-mining problems [11], and hence, any fraud detection system must learn to overcome the problems that are associated with the rarity of positive examples in its learning process. Further complicating this issue, a fraud detection system must deal with overwhelming large volumes of data, typically mostly negative (nonfraudulent) experiences. Creating labeled data for supervised training purposes hence becomes infeasible, and finding relevant real-world training data can be very difficult.

B. Online Learning

Since complete data are not available from the onset (it becomes available gradually over time), we only typically have partial information about the problem at hand [12]. This complicates the learning process as there is a lack of sufficient data for training purposes. Indeed, the detection system may need to approximate or generalize from the limited experiences it has been presented with, but must also be able to quickly adapt as new (potentially conflicting) experiences are observed.

C. Adaptive Adversaries

Another difficult aspect of fraud detection is the perpetual "arms race" that occurs when dealing with intelligent adaptive adversaries who vary their techniques over time [13]. As detection techniques mature, fraud perpetrators also become increasingly sophisticated in both their methods of attack and detection evasion. This means that classifiers may be undermined by their own success; overspecialization is a risk.

D. Concept Drift

In machine learning, concept drift means that the statistical properties of the target variable that the model is trying to predict change over time in unforeseen ways. In other words, a pattern which may indicate nonfraud in time t_1 may be used by fraudsters and indicate fraud in time t_2. This makes the predictions less accurate as time passes. Fraud detection suffers from a continuously changing concept definition [14]. Therefore, the approaches that are used for this purpose should be able to adapt themselves in the presence of drifting concepts over time.

Often the cause of change is hidden. Change can also emerge from the natural evolution of concepts over time or the creation of new concepts, such as the introduction of new formats for email in an email spam detection or change of customers' buying patterns due to inflation.

Another way to categorize concept drift is proposed in [15].

1) *Change in the model:* An observable pattern of activity which was previously considered normal may become a threat.
2) *Change in the number of concepts:* A new type of cyber crime may be introduced (such as phishing, etc.).
3) *Change in the number of features:* New techniques may introduce new features for existing data in order to make classification easier.
4) *Change in the level of noise:* New circumstances may increase or decrease the noise level. For instance, during

the start of each semester, an increase in noise level may occur on a university network due to the increased traffic resulting from students trying to register for courses.

5) *Change in the class distribution:* In network intrusion, discovery of a new vulnerability increases the occurrences of a certain class of attacks and therefore changes the class distribution.
6) *Change in the sample bias:* Conditions under which data are collected affect the sampling process.

E. Noise

Noise is the random perturbations present in the data that leads to variations in observations from the true data signal. The presence of noise in data is almost unavoidable, potentially introduced during any number of stages including data collection, transmission, or processing. Alas, the datasets analyzed and monitored in fraud detection, partly due to their magnitude, have a considerable amount of noise. Unfortunately, the presence of noise makes learning more difficult, typically slowing the rate of learning as the system needs to compensate for false experiences or inaccurate rewards in the case of reinforcement learning. Noise can be in the attribute space or in the label (class). It can be uniformly distributed, or nonuniformly. And finally, it can be randomly generated, or maliciously created by an adversary [16].

Even a small amount of noise is harmful when high accuracy is required. One of the challenges of fraud detection is distinguishing between inconsequential random noise and deliberate fraudulent behavior—fraudulent activity potentially resembling noise due to its atypical appearance [17]. Another challenge is distinguishing between noise and concept drift—a fraud detection system must be robust to noise but sensitive to concept drift.

F. Unequal Misclassification Costs

In fraud detection, misclassification costs (false-positive (FP) and false-negative (FN) error costs) are unequal, are uncertain, can differ from example to example, and can change over time. In some domains, a FN error is more costly than a FP (e.g., intrusion detection), whereas in other domains (e.g., spam detection) the situation is reversed [10]. This adds to the complexity of the learning problem. Another consequence of this property is the absolute need for high accuracy.

G. Fast Processing and Large Volume of Data

In some applications such as network intrusion detection, fraud detection must be done in near real time. In these cases, the speed of processing is critical, thus not affording the fraud detection software the luxury of extended processing times [13]. The large volume of data that fraud detection should process make matters worse.

III. Types of Electronic Fraud

In the last section, we talked about the common properties of fraud detection problems which make this domain a challenging one. In this section, we will discuss the most prevalent types of e-fraud.

The Australian Federal Police defines the term "online fraud" as any type of fraud scheme that uses email, websites, chat rooms, or message boards to present fraudulent solicitations to prospective victims, to conduct fraudulent transactions or to transmit the proceeds of fraud to financial institutions or to others connected with the scheme [18].

In the following sections, the four types of electronic fraud which are most prevalent today are discussed. We also discuss if and how they possess the common properties of e-fraud that are discussed in Section II.

A. Email Spam

Email spam, which is also known as unsolicited commercial email (UCE) or unsolicited bulk email (UBE), is an "unsolicited, unwanted email that is sent indiscriminately, directly, or indirectly, by a sender having no current relationship with the recipient" [19].

Email spam causes numerous problems, the first of which is the cost it imposes on companies and individuals. In 2009, according to a Microsoft security report [20], more than 97% of all emails sent over the Internet were unwanted. Ferris research estimates the cost of spam in 2009 to a total of $130 billion worldwide, a 30% increase over their 2007 estimates [21].

The other problems that are caused by spam are misuse of traffic, storage space, and computational power of the receivers of spam; making users look through and sort out additional email, not only wasting their time and causing loss of work productivity, but also irritating them and, as many claim, violating their privacy rights; and causing legal problems by advertising pornography, pyramid schemes, etc. [22].

With regard to the common properties of e-fraud mentioned before, email spam has most of them.

1) *Experience imbalance*: Based on the level of exposure of an email address on Internet or in the spammers' lists, there is a bias in the ratio of spam emails and ham emails in one's mailbox. According to a Symantec report, in July 2010, the global ratio of spam in email traffic was 88.9% [23].
2) *Online learning*: Although there are training datasets available for spam detectors to learn an initial model offline, there is a need for online learning since spammers tend to change their methods regularly, and parts of the model become obsolete gradually.
3) *Adaptive adversaries:* The reactivity of spammers is a major problem for classification methods, and careful analysis of possible countermeasures is required for any new approach [22]. A simple example of this problem is when spam filters learned that certain words are mostly used in spam emails, spammers decomposed those words into smaller words with spurious punctuations.
4) *Concept drift:* Email spam detection is a challenging problem as the data distribution and concept being learned change over time. An email pattern which used to be classified as legitimate may start to be used by spammers. What makes concept drift more difficult in this case is that

the change is driven by spammers wishing to circumvent spam detection systems [24].

5) *Noise:* The best learning experience by a machine-learning method is achieved when it receives perfectly accurate labels. However, in practice labeling may be far from perfect. In email spam, users give feedback that is often mistaken, inconsistent, or even maliciously inaccurate [25] which can have adverse effects on the performance of an email spam detection system.
6) *Unequal misclassification costs:* Classifying a spam as a legitimate email may just clutter up the inbox by a couple of spam messages, but classifying a legitimate email as spam may be very costly.
7) *Fast processing and large volume of data:* Since receiving emails does not need to be a real-time process and a second delay is usually ignorable, fast processing is not an important issue for email spam detection.

The following approaches to stop, or at least minimize, the amount of spam are listed in [22].

1) *Antispam legislation:* The US CAN-SPAM Act of 2003 allows UCE, but places several restrictions on it. It demands the inclusion of a physical address of the advertiser, an opt-out link in each message, the use of a legitimate return email address, marking messages clearly as advertisements, and prohibits using descriptive subject lines, falsifying header information, harvest email addresses from the Web, and using illegally captured third-party computers to relay messages. The actual compliance with the CAN-SPAM act was low from the very beginning and became even lower later [22].
2) *Modifying email transmission protocols:* This approach attempts to enhance or even substitute the existing standards of email transmission by new, spam-proof variants. One of its most successful implementations is SenderID: The owner of a domain publishes the list of authorized outbound mail servers, thus allowing recipients to check whether a message which pretends to come from a certain domain really originates from there. Released in 2004, this scheme has grown to be quite popular. Almost 40% of legitimate emails today are SenderID-compliant. The other approach is "local changes in email transmission process"; for example, slowing down (assigning a lower priority to) the operations with messages that are likely to be spam [22].
3) *Spam filtering:* It is the most popular approach in stopping email spam. Filtering is the automatic classification of messages into spam and legitimate mail. Existing filtering algorithms are quite effective, often exhibiting an accuracy of above 90% during experimental evaluation. These algorithms can be applied at routers, at the destination mail server, or in the destination mailbox. Filtering on the destination point solves the problems only partially; it prevents end users from wasting their time, but it does not prevent resources misuse because all messages are delivered nevertheless [22]. Filtering algorithms analyze different parts and aspects of email messages: the body or header, individually or in groups, noncontent features (message size), natural language processing (such as removing affixes and ignoring stop words), and image-based filtering (such as OCR, edge detection, color saturation, and average color).

TABLE II
EMAIL SPAM DATASETS

Name	Year	Hams	Spams
Spambase Data Set	1999	2788	1813
Ling-Spam Corpus	2000	2412	481
PU1	2003	618	481
Enron-Spam Dataset	2006	N/A	N/A
TREC 2007 Public Corpus	2007	25220	50199
SpamAssassin Public Mail Corpus	2006	4150	1897

Email spam detection datasets: In order to evaluate their devised methods, researchers require datasets. The dataset required in spam detection should contain a collection of email messages, some of which are marked as spam and the rest as ham (legitimate) messages. Unlike other text categorization tasks, there are only a few publicly available datasets for email spam detection, most of which have only a relatively small number of hams. The reason is that finding and publishing spam messages is easy, but because of privacy issues of the sender and the receiver, publishing hams is mostly infeasible. The other problem with these datasets is that for many of them, spams and hams come from different sources, which may make them biased. Table II summarizes the basic properties of these datasets.

1) *Spambase dataset:* This relatively old dataset from 1999 contains 1813 spam emails either caught by the postmaster or reported by individuals, and 2788 ham emails which come from work and personal emails [26]. In order to avoid privacy issues, the dataset does not provide the email messages, but only 57 features about each of them.
2) *Ling-spam corpus:* The ling-spam corpus consists of 481 spam messages that are received by one of the creators of the corpus and 2412 messages that are sent via the linguist list, a moderated and therefore spam-free list about linguistics. Attachments, HTML tags, and duplicate spam messages that are received on the same day are not included [27]. The corpus was last updated in 2000, and since the spammers methods have improved during the past 10 years, the corpus is now considered mostly outdated.
3) *PU corpora:* The author of ling-spam corpus has also created four other corpora: PUA, PU1, PU2, and PU3, which are collectively called the PU collection. Here, we describe PU1. It is a collection of 480 spam messages and 610 legitimate messages that are received by a particular user after replacing each token (i.e., word, number, punctuation mark, etc.) by a unique number throughout the corpus. Only the earliest five legitimate messages of each sender are retained. Attachments, HTML tags, and duplicate spam messages that are received on the same day are not included [28]. One of the disadvantages of this corpus is the lack of email headers, information that can be used by spam-filtering algorithms in determining the legitimacy of an email.

4) *Enron-spam dataset:* In 2003, the U.S. Federal Energy Regulatory Commission's investigation into Enron declared the email messages of much of the communication between the senior management of the company public. This raw corpus contains about 619 446 internal email messages from 158 users, communicated between 1999 and 2002. The dataset was later purchased by MIT, and some integrity problems were corrected [29]. The cleaned corpus contains 200 399 messages that belong to 158 users, with an average of 757 messages per user [30]. Later on, Metsis *et al.* used these messages as hams and added some spam messages to it to build the Enron-spam dataset [31]. This dataset consists of six datasets, each of which contains ham messages from a single user of the Enron corpus, to which some spams are added (with varying spam ratios for each dataset).
   One advantage of this dataset is that it maintains the order in which the original messages were received. Furthermore, the proportion of hams and spams each user received over time is varied in a similar fashion to real-world experience. This allows a more realistic incremental training of an algorithm [31].
5) *TREC 2007 public corpus:* Created for 2007 Text REtrieval Conference (TREC), the trec07 corpus contains 75 419 messages: 25 220 hams and 50 199 spams. These messages constitute all the messages that are delivered to a particular server between April 8, 2007 and July 6, 2007.
6) *SpamAssassin public mail corpus:* SpamAssassin is an extensible open source email filter that is used to identify spam using a wide range of heuristic tests on mail headers and the text of the email message. As a side project, it provides an email corpus with a spam ratio of 33%, comprising 1897 spam messages and 4150 ham messages.[1] The corpus is updated once in a while with new messages added to it.

B. Phishing

"Phishing is a criminal mechanism employing both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials" [32]. It is done through deceiving computer users to believe that they are dealing with a trustworthy entity, such as a bank. Such attacks are usually done using two media: emails and websites. For example, a phishing email might claim that it is from your bank and because of the recent changes to the underlying system, you need to verify your account details again. It then redirects you to a fake (phishing) website whose URL, look, and feel are almost identical to the legitimate one, prompting you to enter your bank details and other personal information.

Phishing is an example of a social engineering technique which is used to fool users. It exploits the poor usability of current web security technologies [33]. Most methods of phishing use some form of technical deception designed to make a link in an email (and the spoofed website it leads to) appear to belong to the spoofed organization. Misspelled URLs or the use of subdomains are common tricks used by phishers. For example, the URL http://www.hsbc.creditcardupdates.com may seem to belong to the credit card subsystem of HSBC bank. However, it is actually a subdomain (hsbc) belonging to the unknown domain creditcardupdates.com.

The prevalence of phishing attacks has lead to the words "phish" and "phishing" being accepted into the English language. The Oxford English Dictionary defines phishing as "fraud perpetrated on the Internet; spec[ifically] the impersonation of reputable companies in order to induce individuals to reveal personal information, such as passwords and credit card numbers, online" [34].

The Anti-Phishing Working Group (APWG)[2] is an industry association focused on eliminating the identity theft and fraud that result from phishing and email spoofing. The organization provides comprehensive reports on phishing issues periodically. An interesting trend is reported in the APWG's latest report "Phishing Attack Trends Report—First Quarter 2010" [32]: In contrast with most of the previous reports which showed a growth in the number of phishing attacks and websites, the first quarter of 2010 does not show a significant increase in those numbers and, in some cases, a decrease in phishing activity is evident. For instance, the number of unique phishing reports reached a high of 30 577 in March 2010, 25% less than the record in August 2009. Furthermore, the number of total unique phishing websites detected at the end of the March 2010 was 29 879, 47% less than August 2009.

Different attempts have been made to deal with the growing number of reported phishing incidents, including legislation, user training, public awareness, and technical security measures.

The first technique is browser based. In this technique, the browser shows a warning to the user when it suspects an attack. For example, one of the techniques that are used by the phishers is redirecting. The URL http://www.google.com@warez.com may seem to be some section of the Google website. However, what it actually means is logging into the site "warez.com" with the user name "www.google.com"! The Firefox browser detects such tricky requests and shows a warning to the user stating that "Is warez.com the site you want to visit?" The browser also checks the security certificate of the website and reports if it finds anything suspicious.

The main problem with browser warnings is that research shows that users do not pay attention to warnings and messages and press the OK and Accept buttons automatically. This problem is called "Click-thru Syndrome." The authors in [35] analyzed a large set of captured phishing attacks and developed a set of hypotheses about why these strategies might work. Then, they assessed these hypotheses with a study, where 22 participants were shown 20 websites and were asked to determine which ones were fraudulent. It was found that 23% of the participants did not look at browser-based cues such as the address bar, status bar, and the security indicators, leading to incorrect choices 40% of the time. Another usability evaluation was conducted in [36] in order to find out whether the security

[1] http://spamassassin.apache.org/publiccorpus.

[2] http://www.antiphishing.org.

toolbars prevent phishing attacks or not. They conducted two user studies of three security toolbars and other browser security indicators and found them all ineffective at preventing phishing attacks.

The other common approach to address this problem is by eliminating phishing emails in the first place. It is done using phishing email filters which detect as many phishing emails as possible and remove them from inboxes. Similar to email spam detection, this approach relies on machine-learning and natural language processing approaches to classify phishing emails. Some of these approaches are described in [37] and [38].

Not many defense mechanisms against phishing attacks adopt nature-based techniques. However, looking at the literature, we found methods suggested for this purpose which use artificial neural networks [39], [40] or swarm intelligence [41] and, therefore, can be classified under nature-based category. These references will be briefly discussed in Sections IV-A2 and IV-C2.

Finally, let us revisit the common properties of e-fraud and examine them in the context of phishing.

1) *Experience imbalance:* The number of phishing messages/websites is much less than the benign ones. According to a Symantec security report, in July 2010, phishing activity was 1 in 557.5 emails (0.18%) [23].
2) *Online learning:* There are a handful of publicly available phishing datasets for training a phishing detection system, but the changing nature of phishing attacks makes online learning indispensable. Further, the size of the training sets is often too large to be efficiently processed, and the distribution of features that typify malicious URLs is continuously changing [42].
3) *Adaptive adversaries*: As phishing detection tools and methods improve, phishing methods also change in response to find new exploits. According to [43], an important research challenge in phishing detection is to find methods that are able to protect users against adaptive adversaries that operate in real time.
4) *Concept drift:* In their attempts to masquerade attacks, phishers may start using a well-known safe message/website pattern. Hence, a pattern of messages or websites which have correctly been classified as legitimate may now mean an attack. Therefore, a phishing detection system must be able to handle such concept changes [44].
5) *Noise:* Similar to spam emails, phishing data may have considerable noise due to wrong labels, an important practical concern that impacts the applicability of machine-learning techniques for this application [45].
6) *Unequal misclassification costs:* Labeling a legitimate webpage/message as phishing may just give a warning to the user which can be easily ignored, but if a phishing webpage/message is not detected by the system, it may lead to the loss of huge amounts of money.
7) *Fast processing and large volume of data:* For email messages, a second delay in delivery may not be very important, but for browser-based phishing attacks, the processing time may be of more importance.

Phishing datasets: In comparison with spam detection, there are only a few publicly available datasets for phishing detection, with most researchers constructing their own datasets. The authors in [46] constructed a dataset of 2889 emails, 1171 of which are raw phishing emails and 1718 legitimate emails. The authors in [38] used two publicly available datasets to test their implementation: the ham corpora from the SpamAssassin project [47] (both the 2002 and 2003 ham collections, easy and hard, for a total of approximately 6950 nonphishing nonspam emails), and the publicly available phishing corpus [48] (approximately 860 email messages). In addition to being a burden for researchers, constructing and using different datasets makes the comparison between the performance of various approaches very difficult.

C. Network Intrusion

Network intrusion is usually a complicated multistep process which comprises several related actions happening in different parts of the network. The potential result of these actions is usually to change the state of the network in favor of the intruder's intention, for example, accessing unauthorized data [49]. In response, network intrusion detection is the process to monitor events that occurr on a computer system or a network and to analyze them for potential signs of security problems. It is analogous with other monitoring activities such as burglar alarms and security cameras [50]. Intrusion detection deals with the detection of actions that attempt to compromise the confidentiality, integrity, or availability of a resource. It can be performed manually or automatically. Manual intrusion detection might take place by examining log files or other evidence for signs of intrusions, including network traffic. A system that performs automated intrusion detection is called an intrusion detection system (IDS). An IDS can be either host based if it monitors system calls or logs, or network based if it monitors the flow of network packets. Modern IDSs are usually a combination of these two approaches.

There are two primary approaches to analyze events to detect attacks, namely misuse detection and anomaly detection. Misuse detection is based on the extensive knowledge of known attacks and system vulnerabilities provided by a human expert, looking for hackers who attempt to perform these attacks and/or to exploit known vulnerabilities. Although misuse detection can be very accurate in detecting known attacks, it cannot detect unknown and emerging cyber threats [51]. This shortcoming makes them vulnerable to the reactivity of attackers. In other words, when attackers change their behavior in response to detection techniques, these techniques become useless and need major redesign. One solution for this problem would be to use adaptive approaches which are inherently designed to be resilient to small changes in the environment and adapt easily.

On the other hand, anomaly detection is based on the analysis of profiles that represent normal behavior of users, hosts, or network connections. Anomaly detectors characterize normal "legitimate" computer activity using different techniques and then use a variety of measures to detect deviations from defined normal behavior. The major benefit of anomaly detection algorithms is their potential to recognize unforeseen attacks. However, the major limitation is the possibly high false alarm rate. Note that deviations detected by anomaly detection

algorithms may not necessarily represent actual attacks as they may simply be new or unusual but still legitimate network behavior. Anomaly detection techniques fall into the following five groups: statistical methods, rule-based methods, distance-based methods, profiling methods, and model-based approaches [52]. It should be mentioned that many IDSs, such as snort,[3] use both misuse detection and anomaly detection to benefit from their respective advantages [53].

According to [54], the other aspect of an IDS is whether it is passive, active, or proactive. A passive IDS is only responsible for monitoring a system and to inform the administrator once an intrusion occurs or to produce an advance warning. An active IDS, upon detecting an attack, responds to it. Finally, a proactive IDS predicts an intrusion attack before it actually reaches its final stage.

A network intrusion detection system (NIDS) is an IDS that aims to detect malicious activities such as denial of service (DoS) attacks and port scans. A DoS attack or distributed denial of service (DDoS) attack is an attempt to make a computer or network resource unavailable to its intended user. It is usually achieved by sending a huge number of requests from a single computer (DoS) or multiple computers (DDoS). The main targets of these types of attack are companies which heavily rely on online services provided through websites. What makes detection of DDoS attacks difficult is their use of actual source IP address and simulation of normal flows. They work by flooding traffic or using periodically low-rate attack flows on the victim's machine or network. Furthermore, at the beginning of an attack, since traffic fluctuations are kept low, it is difficult to recognize them [55].

The second example of malicious activities in a network is port scanning. It essentially is the cartography of a network and is done by probing a network host for open ports. It is either used by network administrators to verify security policies of their networks or by attackers in order to identify running services on a host with the intention of compromising it for subsequent attacks. Frequent unsuccessful connections from an IP address to different ports may be a sign of a port scan.

Now, let us analyze which of the common properties of e-fraud hold for network intrusion.

1) *Experience imbalance:* Highly imbalanced data distribution is one of the most challenging problem that a network intrusion system faces [56]; some classes of connections (often normal connections) are usually in majority, while some types of attacks are very rare.
2) *Online learning:* Due to the stream-based nature of intrusion detection, it is impractical to store and use all historical data for training [57]. Therefore, a NIDS should use online learning capabilities.
3) *Adaptive adversaries:* Intruders typically try to evade detection and when their methods are discovered by the detection systems, they typically change their tactics.
4) *Concept drift:* New types of attack typically replicate the patterns of a normal connection; hence, what was once considered safe must be regularly checked for anomalous behavior.
5) *Noise:* In network intrusion, noise is created by bad software, bad transmission lines, corrupted DNS servers, local packets which have wrongly been routed outside of the LAN, etc. This noise can deteriorate the performance of a NIDS.
6) *Unequal misclassification costs:* Misclassifying an attack connection as normal is much more dangerous and costly than misclassifying a normal connection as an attack, because in the second case packets will be considered lost and resent which is a standard procedure with computer networks.
7) *Fast processing and large volume of data:* Fast processing is a critical issue in this case, especially when the connection is a video transmission or VOIP.

Network Intrusion Detection Datasets: Privacy concerns and sheer volume of data make the use of captured data from a live network almost impractical. Furthermore, raw data are not of much use without further analysis and labeling [58]. These properties make publicly available network intrusion datasets very rare.

KDD'99: The 1999 KDD cup dataset [59] seems to be the most popularly used dataset in NID research [60]–[68]. This dataset contains 4 898 430 experiences (connections) in the training file and 311 029 in the test file. A smaller version of the training file, containing just 10% subset of the training data at the same distribution as the complete file, is also available. For each file, every "connection" has a vector of 41 features and a label. The label determines if the connection is normal or an attack. There are 22 types of attack in the training file and 39 in the testing file. Every attack type belongs to one of four categories: DOS, R2L, U2R, and probing.

IV. Detection Techniques

In this section, we cover the nature-based techniques and algorithms which have been used for fraud detection and address how each works, its applications, and its advantages and disadvantages.

There are many reasons why computers are very good at e-fraud detection, some of which are related to the high-speed mechanisms required, sharing of information, incremental learning, etc. Therefore, many artificial intelligence techniques have been applied in this domain. Naïve Bayes classifiers have been widely used in email spam detection [69]–[72] as well as in network intrusion detection [63], [73], [74]. Support vector machines (SVMs) have been successfully applied to email spam detection [75], [76], phishing detection [37], network intrusion detection [60], [77], and credit card fraud detection [78], [79]. Visualization has been used to handle email spam detection [80], [81], phishing detection [82], [83], and network intrusion detection [84], [85]. Learning vector quantization [86], fuzzy association rules [87], agent-based IDS [88], and AdaBoost algorithm [89] have been used for network intrusion detection. Hidden Markov model [90], association rules [91],

[3]http://www.snort.org.

and Dempster–Shafer theory [92] have been used for credit card fraud detection.

Statistical methods are also very popular in fraud detection: The authors of [93] give a comprehensive survey of statistical approaches for fraud detection which describes the tools available for statistical fraud detection and the areas in which fraud detection technologies are most used.

However, this survey is looking at another group of techniques called nature-inspired techniques. Many of the mentioned techniques when dealing with changing environments fail to adapt [94]. Nature has been the inspiration of the methods that are discussed here. The reason is that natural systems such as an organ in an animal's body, or a group of animals or insects, at times may seem to behave in a random fashion or be imprecise, but they are robust and resilient. According to [94], this resilience is caused by these factors.

1) Because of redundancy in the system, loss of some parts/elements is not disastrous; parts are easily interchangeable or reproducible.
2) Loose and flexible interconnections between parts of the system make it dynamic and easy to change.
3) Finally, the diversity of parts leads to a diversity of responses to a situation.

The adaptability of natural systems has lured scientists to design computer methods which are inspired by nature and natural systems. The ones which are used in e-fraud detection are discussed next.

A. Artificial Neural Networks

An artificial neural network (ANN), or neural network for short, is a computational model that simulates biological neural networks. It is composed of a large number of neurones that work together to solve specific problems. Usually, an ANN requires training in order to operate (in the case of a supervised ANN). During the training phase, the network changes its weights based on external or internal information that flows through the network. Neural networks can represent both linear and nonlinear relationships and are considered as nonlinear statistical data modeling tools; they can perform tasks that a linear program cannot. Two major applications of ANNs are modeling complex relationships between inputs and outputs, and pattern recognition.

The question is "when is ANN a good choice?" The performance of a neural network relies heavily on the training data. Therefore, when there are not enough data for training, or the quality of data is not desirable (such as noisy data), an ANN is not the right solution. However, if there are plenty of data, or when the problem is difficult to fully understand, an ANN is a good choice. Some application areas for ANNs which yielded good results are pattern recognition [95], [96], medical diagnosis [97], [98], games [99]–[101], and financial applications [102], [103]. We review next some of the papers which have used ANN for fraud detection.

1) Email Spam Detection: The work in [104] describes the design and implementation of a spam filtering optimization system which uses a neural network. Yang and Elfayoumy [105] evaluate the effectiveness of email classifiers based on a feed-forward back-propagation neural network and show that a classifier based on this algorithm provides a relatively high accuracy and sensitivity that makes it competitive to the best known classifiers. However, a neural network system has two major drawbacks: It may be difficult to learn knowledge about minority classes if the set of training data is imbalanced, and the system classification boundaries generated may overlap or not cover some regions of the feature space [106].

2) Phishing Detection: Not too many examples of using ANN can be found in the literature. Exceptions include the following: In [39], ANN is used to detect phishing websites. Based on case studies done in the paper, they find 27 indicators, such as "redirecting pages" and "using IP address," and then group them into six criteria, such as "URL & domain identity" and "page style & contents." Using these groups, the neural network is trained on some publicly available phishing datasets. The trained ANN is then capable of indicating if a website is very legitimate, legitimate, suspicious, phishy, or very phishy. In another work [40], ANN is used as one of the six machine-learning methods to create a classifier ensemble. This classifier ensemble is used to boost the accuracy of heuristics used to identify a phishing email or web page and the results show that it improves the accuracy up to approximately 30%.

3) Network Intrusion Detection: The work in [107] presents a cellular neural network (CNN) templates learning approach based on tabu search to detect network intrusions. Tabu search is a metaheuristic that utilizes memory in order to prevent visiting previously visited solutions (those that are considered "tabu"). They show that their algorithm outperforms both genetic algorithms (GAs) and simulated annealing in terms of computation time and solution quality. Wang *et al.* [108] apply neural networks with work-flow features in IDS in order to improve the detection rate on new attacks and the accuracy of IDSs. This method uses network traffic data to analyze and classify the behaviors of the authorized users and recognize attacks. Their experiments show that this method is effective.

Castellano *et al.* [109] claim neural networks to be one of the most promising methods for intrusion detection, discussing in their paper the huge computation resources required by IDSs as a result of the challenging characteristics of their data sources—such as their large scale, high dimensionality, heterogeneity, and distributed nature. Their solution is a grid-based data-mining approach for an intrusion detection application based on neural networks that can overcome many of these challenges.

B. Evolutionary Algorithms

EAs and GAs are population-based metaheuristic optimization algorithms. An EA uses mechanisms that are inspired by biological evolution: reproduction, mutation, recombination, and selection. Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the environment within which the solutions live. Evolution of the population then takes place after the repeated application of the aforementioned operators. Since EAs do not make any assumption about the underlying environment, they

are considered general algorithms and have been used in a vast range of domains: engineering [110], art [111], biology [112], marketing [113], robotics [114], and medicine [115]. The main problem of these approaches is that they may spend much of the computation time in the encoding and decoding processes. In addition, since GAs typically lack a hill-climbing capacity, they may easily become stuck at local optimum [116]. The versatility of EAs makes them a natural choice for being used in the adversarial and dynamic fraud environments. Some applications of EAs for e-fraud detection are reviewed next.

1) Email Spam Detection: EAs have also been applied in spam filters. Sanpakdee *et al.* [117] used GAs to filter incoming spam mails by generating spam mail prototypes. Dudley *et al.* [118] build a multiobjective EA that evolves its weights for the tests in SpamAssassin according to two independent objectives: simultaneously minimizing the number of FPs (legitimate messages mislabeled as spam), and minimizing the number of FNs (spam messages mislabeled as legitimate).

2) Network Intrusion Detection: Most often, EAs for intrusion detection are used in conjunction with fuzzy logic. Abadeh *et al.* [119] describe a fuzzy genetic-based learning algorithm and discuss its usage to detect intrusion in a computer network. Bridges and Vaughn [120] employ data-mining techniques that utilize fuzzy logic and GAs for intrusion detection. Gong *et al.* [121] present a GA-based approach to network intrusion detection, and a software implementation of the approach. Gomez and Dasgupta [122] propose using GAs to evolve fuzzy rules for detection of intrusions and anomalies in network behavior. Genetic programming [123] has also been widely used for these problems. Crosbie and Spafford [124] view an IDS as multiple functional entities that can be built as autonomous agents. They train these autonomous agents using genetic programming and obtain encouraging results. Orfila *et al.* [125] use genetic programming to produce very lightweight intrusion detection rules and simple patterns that can easily be understood by humans. Suarez-Tangil *et al.* [126] focus on detecting broad attacks (e.g., distributed attacks by botnets) using correlations between events. They present an approach for the generation of security event correlation rules that are based on genetic programming. Mabu *et al.* [127] propose an IDS which is based on fuzzy class-association-rule mining using genetic network (GNP). GNP is an extended EA that represents its solutions using directed graph structures. Combining fuzzy set theory with GNP enables the system to deal with the mixed database that contains both discrete and continuous attributes which is the case of intrusion detection problems.

C. Swarm Intelligence

One interesting phenomenon in nature is the collective behavior of some simple creatures which leads to a collective intelligence. It has led to the creation of a family of artificial intelligence algorithms called swarm intelligence (SI). In SI, a population of unsophisticated agents follow simple rules and have simple communications with each other and their environment, and the resulting collective behavior looks intelligent.

1) Ant Colony Optimization: Computer scientists are using the complex social behaviors of ants to create problem-solving techniques which are able to solve difficult combinatorial optimization problems [4]. The ACO algorithms are based on the ability of ants in finding the shortest paths. Ants do this by leaving pheromones in their trails from their nest to the food and backward. The ants walk randomly to a degree, but generally they are attracted to the path with the most amount of pheromones. The pheromones evaporate after a while, and therefore, the path with the most pheromone will be the shortest path. For more details on the basics of ACO and its applications, see [4].

In the realm of e-fraud detection, ACO has been used in very few instances. The authors in [128] use ACO clustering alongside with dynamic self-organizing maps (DSOM) for network anomaly detection. Their method produces the normal cluster using DSOM and ACO by receiving normal instances during the training phase. Then, during test phase, anomaly data clusters are identified by a normal cluster ratio. The experimental results confirm the efficiency of this approach in detecting unknown intrusions in the case of real network connections.

Another work which uses ACO for anomaly intrusion detection is [129]. The authors' model uses feature extraction algorithms and improves the existing ACO-based clustering algorithms. They evaluate their method against real-world datasets and the results show that it can provide robust and accurate clustering.

ACO has also been used for email spam detection. El-Alfy [130] proposes an ACO-based spam filter. He also compares its performance with three other popular machine-learning techniques: multilayer perceptron, naive Bayes, and ripper classifiers. The results show that this ACO-based filter can give a better accuracy with a considerably smaller rule set.

Another application of ACO is as a feature selection method. For instance, Gao *et al.* [131] use ACO for feature selection alongside SVM for network intrusion detection. Their method is evaluated against KDD cup '99 dataset, and the results show the efficiency of the method in intrusion feature selection and detection.

2) Particle Swarm Optimization: Particle swarm optimization (PSO) is a population-based stochastic optimization method. It is modeled on the basis of the flocking behavior of birds. In the population (swarm), every individual (particle) is a candidate solution. In PSO, each particle is "flown" through the multidimensional search space, adjusting its position in the search space according to its own experience and that of neighboring particles. This way, each particle in addition to the best positions found by itself is using the best position of its neighburs and flying toward an optimum solution while still searching a wide area around the current best solution [132].

In the field of e-fraud detection, PSO has been used mostly in network intrusion detection and very rarely in email spam detection. Xiao *et al.* [133] use the high global search ability of PSO alongside the clustering abilities of K-means method for anomaly intrusion detection. Their experiments on KDD cup '99 dataset shows the outstanding performance of their method.

Similar to ACO, PSO has also been used for feature selection. Srinoy [134] has tackled the intrusion detection problem using PSO and SVM. PSO is used for feature selection, and SVM for anomaly detection. Experimental results show that this method is able to detect not only known attacks, but also new and unknown attacks. Another example is [135] which again uses PSO for feature selection in email spam detection. The authors' experimental results show that PSO can select the most proper discriminative features while eliminating irrelevant ones.

Similar too email spam detection, not too many examples of using PSO can be found in the literature. An exception includes the work in [41] in which the PSO technique is used with an associative classification algorithm for e-banking phishing website detection, and the results are shown to be better than the existing classification algorithms in terms of prediction accuracy and error rate.

D. Artificial Immune Systems

Artificial immune systems (AISs) are adaptive systems that are inspired by the mechanisms of the biological immune system to solve problems [136]. An immune system is a collection of biological processes within an organism that protects against a disease by identifying and killing pathogens and tumor cells. It detects a wide variety of agents, from viruses to parasitic worms, and needs to distinguish them from the organism's own healthy cells and tissues in order to function properly. This concept is very similar to the detection of fraudulent activities and distinguish them from legitimate normal system behavior. The primary advantage of an AIS approach is that it only requires positive examples to learn [137]. Two of the most common applications for AIS are email spam detection [138], [139] and network intrusion detection [140]–[142].

1) Email Spam Detection: AISs are a relatively new approach for spam filtering. Oda and White describe the use of an AIS for protection from spam in [143]. Then, in [144], they test the spam immune system against the publicly available SpamAssassin corpus and extend the original system by looking at several methods of classifying email messages with the detectors produced by the immune system. Finally, in [145], they examine the spam-detecting AIS methods proposed in their two previous papers. Sarafijanovic and Le Boudec [139] present the design and initial evaluation of a new AIS for collaborative spam filtering. The main criticism to the AIS approach is the long time that its calculations require [146] concluding that if the number of entities (email spams) needing processing is large, or timing is important, an AIS will not be a suitable approach.

2) Network Intrusion Detection: Haag *et al.* [147] present an innovative AIS which is integrated with a multiobjective EA. This new distributed IDS design is intended to measure the vector of tradeoff solutions among detectors with regard to two independent objectives: best classification fitness and multiobjective hypervolume size. A new model of self-adaptive network intrusion detection based on the negative selection algorithm (inspired by a process employed by a biological immune system) is presented in [148] to tackle the problem of continuously changing environments in network intrusion detection. Prashanth *et al.* [149] discuss approaches for feature selection and the optimization of parameters, compare different models, as well as discuss other methods to detect anomalies across active networks. A hybrid system is presented in [65] with the aim to combine the advantages of both anomaly detection and misuse detection approaches. Specifically, anomalous network connections are initially detected using an AIS. Connections that are flagged as anomalous are then categorized using a Kohonen self-organizing map (a type of ANN that is trained using unsupervised learning), allowing higher level information (in the form of cluster membership) to be extracted.

E. Learning Classifier Systems

A LCS is a machine-learning technique which combines reinforcement learning, evolutionary computing, and other heuristics to produce an adaptive system capable of learning in dynamic environments [150]. LCS is sometimes referred to as genetic-based machine learning. The successful application of LCSs in a number of related machine-learning and data-mining tasks [151], their online learning capabilities in dynamic environments [152], and their straightforward knowledge representation make LCS a suitable choice for fraud detection systems.

Network intrusion detection: Shafi *et al.* [66] analyze two LCSs (XCS and UCS) on a subset of a publicly available benchmark intrusion detection dataset and introduce a better approach to handle the situation when no rules match an input on the test set, concluding with a recommendation that this be adopted as a standard part of XCS and UCS. They further compare LCS performance with other machine-learning algorithms and conclude that LCSs are a competitive approach to intrusion detection.

Tamee *et al.* combine LCSs with self-organizing maps to build a system for network intrusion detection [153]. They evaluate its performance under an FTP-only dataset and show that the proposed system is able to perform significantly better than the conventional XCS, modified XCS, and 12 standard machine-learning algorithms.

Behdad *et al.* use the continuous version of XCS (called XCSR) to learn and detect network intrusions [154]. Using abstract problems initially, they analyze the behavior of XCSR in the presence of the common properties of the fraud detection problems. Then, they examine the performance of XCSR on the KDD cup '99 network intrusion dataset [59]; their results are comparable with the best known results on that corpus.

V. Optimizing Methods for Fraud Detection Problems

In Section II, some particularly challenging properties that fraud detection systems have to handle were discussed. In addition, in the previous section we examined existing applications of nature-inspired methods to fraud. In this section, we complete this discussion by reviewing techniques that can be used to improve the performance of nature-based fraud detection methods, given the challenging characteristics of fraud.

A. Experience Imbalance

Most AI methods assume that the environments they are learning are well balanced. However in the real world, and specifically fraud detection, data are usually imbalanced. It has been shown that most classification algorithms have difficulty in learning such problems and that the learned model is mostly biased toward the majority class [155]. Some of the methods which are developed to improve the performance of nature-based methods in imbalanced environments are reported next.

1) Sampling Techniques: These methods try to change the class distribution and build artificially balanced training sets by preprocessing the data through oversampling or undersampling [155]. In oversampling, the samples in the minority class are increased to match the samples of the majority class, and in undersampling the samples in the majority class are decreased to match the samples of the minority class. In [156], both mentioned sampling techniques are successfully used alongside the PSO algorithm to handle a problem with imbalanced data.

2) Using Appropriate Evaluation Metrics: Using metrics such as accuracy can be very misleading in imbalanced domains. Consider a scenario in which an environment contains two classes, and the proportion of the minority class to the majority class is 2 to 98. Now, if the algorithm predicts the class of every single input to be the majority class, it will have a 98% accuracy even though it does not predict one minority class input correctly. Of course, typically the minority class has more significance, and hence, this result can be very misleading.

One of the most common evaluation metrics for imbalanced domains is the receiver operating characteristic (ROC) analysis and specifically the "area under an ROC curve" [155]. An example of its usage in EAs is presented in [157] in which EAs are used to handle a problem with imbalanced data. It uses ROC and the area under the ROC curve (AUC) as metrics to measure the performance of classification over imbalanced datasets.

3) One-Class Learning: In one-class learning, classifiers are trained on only one class, and then they are able to distinguish instances belonging to that class from all other possible classes. In this approach, either the majority or the minority class are learned separately [158]. This method has been successfully applied in IDSs [159] and also [160] trains neural networks using one-class learning method for a classification application. Therefore, one can conclude its potential applicability in nature-based methods dealing with fraud detection.

4) Fitness Function Adaptation: The fitness function is an essential part of all EAs and LCSs. It has been shown that it is sensitive to the balance level of classes in data [161]. sUpervised Classifier System (UCS), which is a version of LCSs, uses class-sensitive accuracy [162] to handle class imbalance. This type of accuracy attempts to avoid the bias toward the majority class on imbalanced datasets by penalizing rules covering examples belonging to different classes. Having a class-sensitive rather than instance-sensitive accuracy also helps in the identification of overgeneral classifiers (an overgeneral classifier is a classifier which covers more than one class). However, the system will still suffer from sparsity if minority class instances occur very infrequently with respect to majority class instances.

In another example, Bhowan *et al.* suggest a new fitness function [161] for genetic programming algorithms which is based on the mean square error. It shows to work successfully in an unbalanced environment, without knowing anything about the distribution of classes or any need for preprocessing the data such as oversampling.

5) Adaptive Parameters Used Internally by Algorithms: For some nature-based algorithms, such as PSO and LCS, finding and using the right parameter settings improves the performance of these algorithms significantly. Mostly, the default or common values for the parameter settings are used, but setting the correct value for the parameter settings becomes of more importance under extreme conditions such as in imbalanced environments. The approximate optimized values are usually found by trial and error. However, the ideal way would be to have a dynamic method of changing these values toward the optimized values on the fly. A successful example of this practice is done in [163]. By incorporating an online adaptation algorithm for some of the LCS parameters, the authors have been able to improve the performance of their LCS in the presence of imbalanced data.

B. Adaptive Adversaries

When dealing with adversarial environments, the performance of a classifier deteriorates gradually as the adversary tries to defeat it by changing how it behaves [44]. Adversarial data mining considers the classification problem as a game mechanism between an adaptive adversary and an intelligent classifier. To handle adaptive adversaries, [164] and [165] use game theory in spam detection and IDSs, respectively. Examples of the application of game theory in nature-inspired techniques were not found by the authors. However, it seems a viable approach to try to deal with adaptive adversaries.

C. Concept Drift

Concept drift occurs when a target concept that a learning algorithm is trying to learn changes. It is very common in dynamic environments such as the ones in which fraud usually occurs. Concept drift may affect different components and properties of an environment. Algorithms that deal with such environments should not rely on one-time calculation of the parameters or rule creation, and instead should evaluate rules repeatedly.

There are several approaches to enable nature-based techniques to handle concept drift. According to [24], three of such approaches are briefly described next.

1) *Instance selection*: In instance selection, the instances which are relevant to the current concept are selected. The windowing technique is the most popular example. It only uses the most recent instances or rules learnt to deal with the current situation. Koychev [166] suggests using a time window which is implemented as a time-based gradual "forgetting" function. He shows that this method improves prediction accuracy on drifting concepts. Lazarescu *et al.* [167] propose using a multiple window incremental learning algorithm that uses competing windows to interpret the data and hence detect concept

drift. The advantage of using multiple windows instead of one window is that it allows the system to progressively adapt and predict the change, and therefore deal more effectively with different types of drift.

2) *Instance weighting*: Unlike "instance selection" which selects and only uses the most relevant instances, "instance weighting" gives weight to instances that are based on their relevance and age, and then uses the ones with the highest weight.
3) *Ensemble learning*: Ensemble learners maintain a set of learners (agents) and combine their decisions for final decision making. Usually applying the same algorithm to one dataset produces the same classifiers. Hence, in order to create different classifiers, either an algorithm is trained on different datasets or different algorithms are used [168].

In addition to these, it is important to recognize long-term trends that only occur once in a while (e.g., seasonal trends that occur in certain months of the year) and treat them accordingly. Finally, an important issue in dealing with concept drift is being able to distinguish between concept drift and noise, both of which may manifest with similar effects on the environment, but can represent very different underlying properties of the model.

D. Noise

Noise is the random error or variance in a measured variable [169]. Due to the magnitude of fraud detection datasets, noisy data or statistical error is unavoidable. The solution for this problem is either using techniques which neutralize noise, improving algorithms [170], or preprocessing data [169], [171].

A common technique for this problem is "resampling." When dealing with noisy environments, relying on only one sample is nonsensical. The reason is that a sample may contain noise and therefore be inaccurate. By resampling, that is taking a number of samples and then using the arithmetic mean of the samples, the effect of noise can be greatly reduced. In environments where different parts have different levels of noise, it is desirable to have dynamic resampling, which uses different (dynamic) resampling rates for different points of the environment [172]. For instance, the authors in [173] improve steady-state GAs in dealing with noisy fitness functions by introducing a new dynamic resampling technique.

E. Unequal Misclassification Costs

In fraud detection problems, the cost of a FN is usually not equal to that of a FP; often the former is much costlier. Therefore, a good fraud detection system will have the ability to bias its classifier toward making fewer errors of the costlier type, even if it means making more mistakes of the other (cheaper) type.

In order to take into account the asymmetry in the misclassification costs, performance can be measured using cost-sensitive metrics. Two popular ones are weighted accuracy and total cost ratio. In the case of spam detection, if S and L are, respectively, the sets of spam and legitimate messages and a FP is considered to be λ times more costly than a FN, then the weighted accuracy and the total cost ratio are calculated as follows [174]:

1) Weighted Accuracy: $\text{Acc}_w = \frac{|\text{TP}|+\lambda|\text{TN}|}{|S|+\lambda|L|}$
2) Total Cost Ratio: $\text{TCR} = \frac{|S|}{\lambda|\text{FP}|+|\text{FN}|}$.

Another approach is weighting the data space by changing the distribution of the training set with regard to misclassification costs so that the distribution is biased toward the costly classes [175]. In other words, the classifier will receive a higher number of instances from more costly class.

Finally, in a nature-based technique, such as LCS, which contains a reinforcement learning component, the reward that the reinforcement learning component returns can be tuned in a way that the misclassification of one class is costlier than the other class(es).

F. Fast Processing and Large Volume of Data

To improve the processing speed of nature-inspired techniques and in response to the large volume of data typically inherent in fraud detection problems, the following methods have been employed.

1) Using estimation techniques instead of calculating exact values [176], [177] can be used in any part of the nature-based techniques, such as fitness value calculation in EAs.
2) *Parallelization using graphics processing units (GPUs):* The general-purpose graphics processing units (GPGPU) in modern computers are so powerful that researchers have started to use them in high performance computing. An important characteristic of nature-based techniques is that they demand relatively high resources. It makes the usage of the power of GPUs very rewarding for systems that use nature-based techniques. A recent example of this trend is the work in [178] in which by parallelizing the most time-consuming components of the BioHEL algorithm—a version of LCS—, it achieves speedups up to 58.1×. The authors use NVIDIA's parallel technology using the CUDA library. The methodology proposed in the research can be easily extended to any evolutionary learning system.
3) *Dimensionality reduction and feature selection:* When dealing with high-dimensional data, time and computational requirements grow exponentially. Dimensionality reduction techniques such as principal component analysis (PCA) [179] will help in such situations by transforming data into a new space in which the first few features will usually represent the majority of information. Behdad *et al.* [180] use PCA alongside XCSR (a version of LCSs) to enable XCSR to handle high-dimensional problems in a reasonable time using reasonable resources. Another technique is feature selection which can identify the most relevant and discriminating features of the original data. For instance, the PSO has been used in [135] to identify the most important and relevant features in email spam detection.
4) To improve the efficiency of LCSs, Bacardit and Krasnogor [181] suggest a new representation for continuous attributes. Their method is based on the fact that in large datasets, it is often the case that only a very small fraction

TABLE III
TYPES OF E-FRAUD VERSUS DETECTION APPROACHES

Type of E-Fraud	ANN	EA	SI	AIS	LCS
Email Spam	[104], [105]	[117], [118]	[130], [135]	[143], [144], [145], [139]	
Phishing	[39], [40]		[41]		
Network Intrusion	[107], [108]	[119], [120], [121], [122], [124], [125], [126], [127]	[128], [129], [131], [133], [134]	[147], [148], [149], [65]	[66], [153], [154]

of the attributes of a domain are expressed at the same time in a rule. Automatically discovering these key attributes and keeping track of only them contributes to substantial speed ups.

VI. CONCLUSIONS AND FUTURE WORKS

Electronic fraud presents several significant challenges to learning algorithms. In Section V, we have seen that there are techniques and methods available to meet these challenges.

Table III summarizes representative techniques in the literature that are used for each fraud detection problem. According to this table, email spam detection uses most of the techniques except LCS. For phishing detection, it usually utilizes a list of features to score the likelihood of a message being genuine or a phish. ANN optimizes the weights of these features and is the most popular technique in phishing detection as illustrated in Table III. Nature-inspired techniques have been widely used for network intrusion detection, particularly EA and SI. Finally, in regard to the empty cells, they indicate either opportunities for future contribution or incapabilities of the technique to deal with the respective fraud detection problems. For instance, in the case of LCS, although it has been applied successfully in network intrusion detection [66], [153], [154], it has not been tried in the other fields.

A. Emerging Types of Electronic Fraud

As new technologies are developed, new opportunities for fraud emerge. In this section, some recent types of e-fraud which have have been growing in frequency over the past few years are examined.

1) Internet Auction Fraud: An Internet auction fraud occurs when a bidder wins an auction on the Internet and pays for it, but does not receive the item or the item received is not what was described in the auction. Internet auction fraud is one of the main sources of Internet fraud, with estimates of incidence from 64% to 87% of all Internet fraud [182].

2) Click Fraud: Pay per click advertising is an arrangement in which operators of a website display clickable links from advertisers in exchange for a charge per click. Click fraud is the practice of deceptively clicking on search advertisements with the intention of either increasing third-party website revenue or exhausting an advertiser's budget [183]. It may be done by a person, automated script, or computer program that imitates a legitimate user of a web browser clicking on an advertisement. A newly uncovered form of click fraud is click laundering in which technical measures are used to make invalid "ad-clicks" appear to originate from legitimate sources. It is analogous to money laundering in which the origin of illegal profits is disguised as legitimate [184].

3) Mule Recruitment: Mule recruitment is an attempt to get a person (the "money mule") to receive stolen funds using his or her bank account and then transfer those funds to criminals overseas [18]. This is the method used by many phishers for money laundering. Recruiting "money mules" is usually done through an email which explains that the mule will receive money from an account (a compromised phishing victim account) from which they will take a cut, and then must forward the rest of the money to a third party using nonrevocable transactions (Western Union transfers for instance). When the police finds out about the fraudulent transfers, the transfer is reversed and the mule may end up paying the money back to the original phishing victim [185].

4) Spam Blog: A spam blog, which is also known as splog, is a blog which is designed for the purpose of promoting affiliated websites, increasing their search engine rankings or to just sell advertisements. The author in [186] uses a SVM-based approach on local and link-based features to detect splogs.

5) Review Spam: It is common practice for Internet shopping websites to have comment/review systems. This way, the customer can write about the product they have bought and talk about its pros and cons. It is both good for a potential buyer to get to know more about the product and also the manufacturer to improve its products. The problem is that these services can also be misused. Some product owners write unjustified positive reviews about their products as bogus users, and some write unfair negative reviews about their competitor's products. For example, the authors in [187] analyzed 5.8 million reviews from 2.14 million reviewers from amazon.com and show that review spam is quite widespread. They also detail in [188] a detection method using duplicate finding and 2-class classification (spam and nonspam) which works effectively in detecting such spam.

6) Web Vandalism: Vandalism is defined as intentional destruction of property. Web vandalism includes activities such as deliberating defacing a webpage or a website. For instance, a politically motivated hacker may change key messages on a website for political gain or embarrassment. Web vandalism is common in Wikipedia, manifesting in the form of destructive article revisions. Vandalism detection tries to detect the blatant unproductive edits among all revisions [189].

7) Plagiarism: Merriam-Webster Dictionary defines "plagiarise" as "1) to steal and pass off (the ideas or words of another) as one's own; 2) use (another's production) without crediting the

source; 3) to commit literary theft; present as new and original an idea or product derived from an existing source." The Internet has made life very easy for plagiarizers: just by searching for any word or phrase one can come up with thousands of texts and materials which can be readily copied and pasted in a new document and published as a new or original work. The notable victims of this type of fraud are content generators such as weblogers, news websites, and academics. Plagiarism detection is usually based on finding similarities between different documents. Some of the similarity metrics in this regard are discussed in [190].

B. Opportunities

Nature-inspired techniques have few unique characteristics which make them a valuable technique to be used in e-fraud detection: They are versatile. It is not required to have a complete knowledge of the domain they are used for. They do not need tuning. In many cases of e-fraud detection problems, we do not have complete information of the environment. This does not cause any difficulty for nature-inspired techniques. In addition, these techniques are very robust. They can adapt as the environment changes. This is particularly very essential in the dynamic world of e-fraud detection. We have seen in a number of instances, nature-inspired techniques being successfully used in detecting and preventing electronic fraud. For instance, most notably in the instance of network intrusion detection, AISs [140] are used in a large scale. The characteristics of nature-inspired techniques mentioned previously suggest that they have further potential, especially on the newer types of e-fraud.

In addition, the large volumes of data that must be handled in electronic domains and the occurrence of adaptive adversaries mean that there will always be the threat of newer types of increasingly complex fraud. Where some techniques such as EAs [125] and ANNs [105] may expect continued success on some instances of fraud, new challenges may arise: the variable nature of splogs will require more robust variations of AI algorithms [191]; network intrusion increasingly requires researchers to model temporal aspects of attacks in which a series of actions that are innocuous by themselves become malicious when presented in a coordinated way (e.g., port scans); click fraud also presents the challenge of modeling temporal aspects of potentially fraudulent entities.

To be truly effective, automated fraud detection techniques will need to become increasingly complex in the way they model agents and entities. For example, there is true potential to enhance rule-based systems such as genetic programming and LCSs to detect suspect behaviors rather than suspect actions. Another significant challenge is providing detection techniques that are responsive in a rapidly changing domain. In addition, tagging corpuses for training is an expensive task; therefore, the use of data mining, clustering, and semisupervised learning techniques will be a crucial aspect of e-fraud detection in the future.

REFERENCES

[1] Australian Bureau of Statistics. (2008). "Personal fraud, 2007,' Australian Bureau of Statistics, Canberra, Australia, Tech. Rep. 4528.0. [Online]. Available:http://www.abs.gov.au/ausstats/abs@.nsf/cat/4528.0

[2] Internet Crime Complaint Center. (2010). "IC3 2009 annual report," [Online]. Available:http://www.ic3.gov/media/annualreport/2009_IC3Report.pdf

[3] D. Kirk, "Fighting fraud," *J. Crim. Law*, vol. 72, no. 5, pp. 335–337, 2008.

[4] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MI: Bradford Company, 2004.

[5] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. New York: Springer-Verlag, 2003.

[6] P. J. Fleming and R. C. Purshouse, "Evolutionary algorithms in control systems engineering: A survey," *Contr. Eng. Pract.*, vol. 10, no. 11, pp. 1223–1241, 2002.

[7] I. Nikolos, K. Valavanis, N. Tsourveloudis, and A. Kostaras, "Evolutionary algorithm based offline/online path planner for UAV navigation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 6, pp. 898–912, Dec. 2003.

[8] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.

[9] S. W. Wilson, "Classifier fitness based on accuracy," *Evol. Comput.*, vol. 3, no. 2, pp. 149–175, 1995.

[10] C. Phua, V. C. S. Lee, K. Smith-Miles, and R. W. Gayler, "A comprehensive survey of data mining-based fraud detection research," *CoRR*, vol. abs/1009.6119, pp. 1–14, 2010.

[11] G. M. Weiss, "Mining with rarity: A unifying framework," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 7–19, 2004.

[12] V. Vatsa, S. Sural, and A. Majumdar, "A game-theoretic approach to credit card fraud detection," in *Information Systems Securit*, (Lecture Notes in Computer Science vol. 3803). New York: Springer, 2005, pp. 263–276.

[13] T. Fawcett, I. Haimowitz, F. Provost, and S. Stolfo, "AI approaches to fraud detection and risk management," *AI Mag.*, vol. 19, no. 2, pp. 107–108, 1998.

[14] K. Shafi and H. A. Abbass, "Biologically-inspired complex adaptive systems approaches to network intrusion detection," *Inf. Security Tech. Rep.*, vol. 12, no. 4, pp. 209–217, 2007.

[15] H. A. Abbass, J. Bacardit, M. V. Butz, and X. Llorà, "Online adaptation in learning classifier systems: Stream data mining," Illinois Genetic Algorithms Lab., Univ. Illinois Urbana-Champaign, Tech. Rep. 2004031, 2004.

[16] M. Kearns and M. Li, "Learning in the presence of malicious errors," in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 267–280.

[17] A. Baglioni and U. Cherubini, "Accounting fraud and the pricing of corporate liabilities: Structural models with garbling," in *SSRN eLibrary*, 2007.

[18] Australian Federal Police. (2010). "Internet fraud and scams," [Online]. Available:http: //www.afp.gov.au/policing/e-crime/internet-fraud-and-scams.aspx

[19] G. V. Cormack and T. Lynam, "TREC 2005 spam track overview," in *Proc. 14th Text Retrieval Conf.*, 2005, pp. 1–17.

[20] BBC News. (2009). "Spam overwhelms e-mail messages," BBC, London, U.K. [Online]. Available:http: //news.bbc.co.uk/2/hi/technology/7988579.stm

[21] R. Jennings. (2009). "Cost of spam is flattening—Our 2009 predictions," [Online]. Available:http: //email-museum.com/2009/01/28/cost-of-spam-is-flattening-our-2009- predictions/

[22] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artif. Intell. Rev.*, vol. 29, no. 1, pp. 63–92, 2008.

[23] Symantec Press Release. (2010). "Symantec announces July 2010 MessageLabs intelligence report," Symantec, CA. [Online]. Available:http: //www.symantec.com/about/news/release/article.jsp?prid=20100722_0 1

[24] S. J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle, "A case-based technique for tracking concept drift in spam filtering," *Knowl.-Based Syst.*, vol. 18, no. 4–5, pp. 187–195, 2005.

[25] D. Sculley and G. Cormack, "Filtering email spam in the presence of noisy user feedback," in *Proc. 5th Email Anti-Spam Conf.*, 2008, pp. 1–10.

[26] M. Hopkins, E. Reeber, G. Forman, and J. Suermondt, "Spambase data set," (1999). [Online]. Available:http: //archive.ics.uci.edu/ml/datasets/Spambase

[27] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos, "An evaluation of naive Bayesian anti-spam filtering," in *Proc. Workshop Mach. Learn. New Inf. Age*, 2000, pp. 9–17.

[28] A. Bratko, B. Filipič, G. Cormack, T. Lynam, and B. Zupan, "Spam filtering using statistical data compression models," *J. Mach. Learn. Res.*, vol. 7, pp. 2673–2698, 2006.

[29] CALO Project, "Enron email dataset," (2009). [Online]. Available:http: //www-2.cs.cmu.edu/enron/

[30] B. Klimt and Y. Yang, "Introducing the enron corpus," in *Proc. 1st email Anti-Spam Conf.*, 2004, pp. 1–2.

[31] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive Bayes—Which naive bayes?," in *Proc. 3rd Email Anti-Spam Conf.*, 2006, pp. 125–134.

[32] Anti-Phishing Working Group. (2010). "Phishing attack trends report—First quarter 2010," [Online]. Available:http: //www.antiphishing.org/reports/apwg_report_Q1_2010.pdf

[33] A. Josang, B. AlFayyadh, T. Grandison, M. AlZomai, and J. McNamara, "Security usability principles for vulnerability analysis and risk assessment," in *Proc. 23rd Annu. Comput. Security Appl. Conf.*, 2007, pp. 269–278.

[34] OED Online. (1989). "phishing, n," [Online]. Available:http: //dictionary.oed.com/cgi/entry/30004304

[35] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2006, pp. 581–590.

[36] M. Wu, R. C. Miller, and S. L. Garfinkel, "Do security toolbars actually prevent phishing attacks?," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2006, pp. 601–610.

[37] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya, "Phishing E-mail detection based on structural properties," in *Proc. 1st Annu. Symp. Inf. Assur.: Intrus. Detect. Prevent.*, 2006, pp. 2–8.

[38] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proc. 16th Int. World Wide Web Conf.*, 2007, pp. 649–656.

[39] A. Martin, N. B. Anutthamaa, M. Sathyavathy, M. M. S. Francois, and V. P. Venkatesan, "A framework for predicting phishing websites using neural networks," *Int. J. Comput. Sci. Issues*, vol. 8, pp. 330–336, 2011.

[40] N. Sanglerdsinlapachai and A. Rungsawang, "Web phishing detection using classifier ensemble," in *Proc. 12th Int. Inf. Integr. Web-Based Appl. Serv. Conf.*, 2010, pp. 210–215.

[41] R. Damodaram and M. Valarmathi, "Phishing website detection using particle swarm optimization," *Int. J. Comput. Sci. Security*, vol. 5, no. 5, pp. 477–490, 2011.

[42] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious URLs: An application of large-scale online learning," in *Proc. 26th Annu. Int. Mach. Learn., Conf.*, 2009, pp. 681–688.

[43] R. Oppliger and S. Gajek, "Effective protection against phishing and web spoofing," in *Communications and Multimedia Security*, J. Dittmann, S. Katzenbeisser, and A. Uhl, Eds., (Lecture Notes in Computer Science vol. 3677) Berlin/Heidelberg, Germany: Springer-Verlag, 2005, pp. 32–41.

[44] G. L'Huillier, R. Weber, and N. Figueroa, "Online phishing classification using adversarial data mining and signaling games," in *Proc. ACM SIGKDD Workshop Cyber Security Intell. Informat.*, 2009, pp. 33–42.

[45] A. Kolcz and G. V. Cormack, "Genre-based decomposition of email class noise," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2009, pp. 427–436.

[46] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. Anti-Phishing Working Groups 2nd Annu. eCrime Researchers Summit*, 2007, pp. 60–69.

[47] J. Mason. (2005). "The apache spamassassin public corpus," [Online]. Available:http: //spamassassin.apache.org/publiccorpus

[48] J. Nazario. (2005). "Phishing corpus," [Online]. Available:http: // monkey.org/jose/wiki/doku.php?id=PhishingCorpus

[49] S. Zhang, J. Li, X. Chen, and L. Fan, "Building network attack graph for alert causal correlation," *Comput. Security*, vol. 27, no. 5–6, pp. 188–196, 2008.

[50] R. Bace, *Intrusion Detection*. Indianapolis, IN: Sams, 2000.

[51] V. Kumar and J. Srivastava, A. Lazarevic, *Managing Cyber Threats: Issues, Approaches, and Challenges*. New York: Springer-Verlag, 2005.

[52] A. Lazarevic, V. Kumar, and J. Srivastava, "Intrusion detection: A survey," in *Proc. Manag. Cyber Threats*, 2005, pp. 19–78.

[53] A. Seleznyov and S. Puuronen, "HIDSUR: A hybrid intrusion detection system based on real-time user recognition," in *Proc. 11th Int. Workshop Database Expert Syst. Appl.*, 2000, pp. 41–45.

[54] P. Kabiri and A. Ghorbani, "Research on intrusion detection and response: A survey," *Int. J. Netw. Security*, vol. 1, no. 2, pp. 84–102, 2005.

[55] J. Cheng, J. Yin, Y. Liu, Z. Cai, and C. Wu, "Detecting distributed denial of service attack based on multi-feature fusion," in *Security Technology*, (Communications in Computer and Information Science, vol. 58). Berlin/Heidelberg, Germany: Springer-Verlag, 2009, pp. 132–139.

[56] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Appl. Soft Comput.*, vol. 10, no. 1, pp. 1–35, 2010.

[57] M. M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham, "Peer to peer botnet detection for cyber-security: A data mining approach," in *Proc. 4th Annu. Workshop Cyber Security Inf. Intell. Res.*, 2008, pp. 1–2.

[58] H. G. Kayacık and N. Zincir-Heywood, "Analysis of three intrusion detection system benchmark datasets using machine learning algorithms," in *Intelligence and Security Informatics*, P. Kantor, G. Muresan, F. Roberts, D. D. Zeng, F.-Y. Wang, H. Chen, and R. C. Merkle, Eds., (Lecture Notes in Computer Science, vol. 3495) Berlin/Heidelberg, Germany: Springer-Verlag, 2005, pp. 362–367.

[59] S. Stolfo *et al.* (2007). "KDD cup 1999 dataset," [Online]. Available:http: //archive.ics.uci.edu/ml/datasets/KDD+Cup+1999+Data

[60] D. S. Kim, H.-N. Nguyen, and J. S. Park, "Genetic algorithm to improve SVM based network intrusion detection system," in *Proc. 19th Int. Conf. Adv. Inf. Network. Appl.*, 2005, vol. 2, pp. 155–158.

[61] W. Ma, D. Tran, and D. Sharma, "A study on the feature selection of network traffic for intrusion detection purpose," in *Proc. Conf. Intell. Security Informat.*, 2008, pp. 245–247.

[62] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," in *Proc. 28th Australasian Conf. Comput. Sci*, 2005, pp. 333–342.

[63] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naïve Bayes vs decision trees in intrusion detection systems," in *Proc. 2004 ACM Symp. Appl. Comput.*, 2004, pp. 420–424.

[64] V. Engen, J. Vincent, and K. Phalp, "Enhancing network based intrusion detection for imbalanced data," *Int. J. Knowl.-Based Intell. Eng. Syst.*, vol. 12, no. 5–6, pp. 357–367, 2008.

[65] S. T. Powers and J. He, "A hybrid artificial immune system and self organising map for network intrusion detection," *Inf. Sci.*, vol. 178, no. 15, pp. 3024–3042, 2008.

[66] K. Shafi, T. Kovacs, H. A. Abbass, and W. Zhu, "Intrusion detection with evolutionary learning classifier systems," *Nat. Comput.*, vol. 8, no. 1, pp. 3–27, 2009.

[67] K. Shafi, H. Abbass, and W. Zhu, "An adaptive rule-based intrusion detection architecture," presented at the Security Technol. Conf., 5th Homeland Security Summit, Canberra, Australia, 2006.

[68] K. Shafi, H. A. Abbass, and W. Zhu, "The role of early stopping and population size in xcs for intrusion detection," in *Simulated Evolution and Learning*, (Lecture Notes in Computer Science vol. 4247). Berlin, Germany: Springer, 2006, pp. 50–57.

[69] Z. Yang, X. Nie, W. Xu, and J. Guo, "An approach to spam detection by naïve Bayes ensemble based on decision induction," in *Proc. 6th Int. Conf. Intell. Syst. Design Appl.*, 2006, pp. 861–866.

[70] A. K. Seewald, "An evaluation of naïve Bayes variants in content-based learning for spam filtering," *Intell. Data Anal.*, vol. 11, no. 5, pp. 497–524, 2007.

[71] H. Zhang and D. Li, "Naïve Bayes text classifier," in *Proc. IEEE Int. Conf. Gran. Comput.*, 2007, pp. 708–708.

[72] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "Bayesian additive regression trees-based spam detection for enhanced email privacy," in *Proc. 3rd Int. Avail., Reliabil. Security Conf.*, 2008, pp. 1044–1051.

[73] K.-C. Khor, C.-Y. Ting, and S.-P. Amnuaisuk, "A probabilistic approach for network intrusion detection," in *Proc. 2008 2nd Asia Int. Conf. Modell. Simul.*, 2008, pp. 463–468.

[74] E. Lauria and G. Tayi, "Statistical machine learning for network intrusion detection: A data quality perspective," *Int. J. Serv. Sci.*, vol. 1, no. 2, pp. 179–195, 2008.

[75] C.-Y. Chiu and Y.-T. Huang, "Integration of support vector machine with naïve Bayesian classifier for spam classification," in *Proc. 4th Int. Fuzzy Syst. Knowl. Discov. Conf.*, 2007, pp. 618–622.

[76] D. Sculley and G. M. Wachman, "Relaxed online SVMs for spam filtering," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Development Inf. Retrieval*, 2007, pp. 415–422.

[77] X. Xu and X. Wang, "An adaptive network intrusion detection method based on PCA support vector machines," in *Advanced Data Minning and Application*, (Lecture Notes in Computer Science, vol. 3584). Berlin, Germany: Springer, 2005, pp. 696–703.

[78] R. Chen, T. Chen, Y. Chien, and Y. Yang, "Novel questionnaire-responded transaction approach with SVM for credit card fraud detection," in *Advances in Neural Network*, (Lecture Notes in Computer Science, vol. 3497). Berlin: Springer, 2005, pp. 916–921.

[79] R. Chen, T. Chen, and C. Lin, "A new binary support vector system for increasing detection rate of credit card fraud," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 20, no. 2, pp. 227–240, 2006.

[80] C.-T. Wu, K.-T. Cheng, Q. Zhu, and Y.-L. Wu, "Using visual features for anti-spam filtering," in *Proc. Int. Conf. Image Process.*, 2005, vol. 3, pp. 509–512.

[81] B. Mehta, S. Nangia, M. Gupta, and W. Nejdl, "Detecting image spam using visual features and near duplicate detection," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 497–506.

[82] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng, "Detection of phishing webpages based on visual similarity," in *Proc. 14th Int. Conf. World Wide Web*, 2005, pp. 1060–1061.

[83] A. Y. Fu, L. Wenyin, and X. Deng, "Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD)," *IEEE Trans. Dependable Secure Comput.*, vol. 3, no. 4, pp. 301–311, Oct.–Dec. 2006.

[84] J. Zhang, G. Yang, L. Lu, M. Huang, and M. Che, "A novel visualization method for detecting DDoS network attacks," *Vis. Inf. Commun.*, pp. 185–194, 2009.

[85] T. Itoh, H. Takakura, A. Sawada, and K. Koyamada, "Hierarchical visualization of network intrusion detection data," *IEEE Comput. Graph. Appl.*, vol. 26, no. 2, pp. 40–47, 2006.

[86] Y. Degang, C. Guo, W. Hui, and L. Xiaofeng, "Learning vector quantization neural network method for network intrusion detection," *Wuhan Univ. J. Nat. Sci.*, vol. 12, no. 1, pp. 147–150, 2007.

[87] M.-Y. Su, S.-C. Yeh, K.-C. Chang, and H.-F. Wei, "Using incremental mining to generate fuzzy rules for real-time network intrusion detection systems," in *Proc. 22nd Int. Adv. Inf. Network. Appl. Conf.*, 2008, pp. 50–55.

[88] M. Rehak, M. Pechoucek, P. Celeda, V. Krmicek, M. Grill, and K. Bartos, "Multi-agent approach to network intrusion detection," in *Proc. 7th Int. Joint Auton. Agents Multiagent Syst. Conf.*, 2008, pp. 1695–1696.

[89] W. Hu, W. Hu, and S. Maybank, "Adaboost-based algorithm for network intrusion detection," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 577–583, Apr. 2008.

[90] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit card fraud detection using hidden Markov model," *IEEE Trans. Dependable Secure Comput.*, vol. 5, no. 1, pp. 37–48, Jan.–Mar. 2008.

[91] D. Sanchez, M. Vila, L. Cerda, and J. Serrano, "Association rules applied to credit card fraud detection," *Expert Syst. Appl.*, vol. 36, no. 2, Part 2, pp. 3630–3640, 2009.

[92] S. Panigrahi, A. Kundu, S. Sural, and A. Majumdar, "Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning," *Inf. Fusion*, vol. 10, no. 4, pp. 354–363, 2009.

[93] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," *Statist. Sci.*, vol. 17, no. 3, pp. 235–249, 2002.

[94] P. Marrow, "Nature-inspired computing technology and applications," *BT Technol. J.*, vol. 18, pp. 13–23, 2000.

[95] X. Yan-hong, Z. Ze, L. Kun, and Z. Guan-ying, "Fuzzy neural networks pattern recognition method and its application in ultrasonic detection for bonding defect of thin composite materials," in *Proc. IEEE Int. Conf. Autom. Logist.*, 2009, pp. 1345–1349.

[96] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[97] Y. Lin, C. Fan, C. Huang, and C. Fan, "Construct an approximation decision model of medical record by neural networks—The ophthalmology department as an example," in *New Advances in Intelligent Decision Technologies: Results of the First KES International Symposium IDT 2009*. New York: Springer-Verlag, 2009, pp. 467–480.

[98] D. Gil, M. Johnsson, J. M. G. Chamizo, A. S. Paya, and D. R. Fernandez, "Application of artificial neural networks in the diagnosis of urological dysfunctions," *Expert Syst. Appl.*, vol. 36, no. 3, Part 2, pp. 5754–5760, 2009.

[99] T. Schaul and J. Schmidhuber, "Scalable neural networks for board games," in *Proc. Int. Artif. Neural Netw. Conf.*, 2008, pp. 1005–1014.

[100] T. Schaul and J. Schmidhuber, "A scalable neural network architecture for board games," in *Proc. IEEE Symp. Comput. Intell. Game*, 2008, pp. 357–364.

[101] D. Michulke and M. Thielscher, "Neural Networks for state evaluation in general game playing," in *Proc. Eur. Conf. Mach. Learning and Knowl. Discovery in Databases: Part II*, 2009, pp. 95–110.

[102] C. Pei-Chann Chang, C. Fan, J. Lin, and C. Lai, "An ensemble of neural networks for stock trading decision making," in *Emerging Intelligent Computing Technology and Applications. With Aspects of Artificial Intelligence*, (Lecture Notes in Computer Science vol. 5755). New York: Springer-Verlag, 2009, pp. 1–10.

[103] A. J. Hussain, A. Knowles, P. J. Lisboa, and W. El-Deredy, "Financial time series prediction using polynomial pipelined neural networks," *Expert Syst. Appl.*, vol. 35, no. 3, pp. 1186–1199, 2008.

[104] C. Zhan, F. Zhang, and M. Zheng, "Design and implementation of an optimization system of spam filter rule based on neural network," in *Proc. Int. Commun., Circuits Syst. Conf.*, 2007, pp. 882–886.

[105] Y. Yang and S. A. Elfayoumy, "Anti-spam filtering using neural networks and Baysian classifiers," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, 2007, pp. 272–278.

[106] G. Ou and Y. L. Murphey, "Multi-class pattern classification using neural networks," *Pattern Recognit.*, vol. 40, no. 1, pp. 4–18, 2007.

[107] Z. Yang, A. Karahoca, N. Yang, and N. Aydin, "Network intrusion detection by using cellular neural network with tabu search," in *Proc. 2008 Bio-Inspired, Learn. Intell. Syst. Security*, 2008, pp. 64–68.

[108] Y. Wang, D. Gu, W. Li, H. Li, and J. Li, "Network intrusion detection with workflow feature definition using bp neural network," in *Proc. 6th Int. Symp. Neural Netw. Adv. Neural Netw.*, 2009, pp. 60–67.

[109] M. Castellano, G. Mastronardi, and G. Tarricone, "Intrusion detection using neural networks: A grid computing based data mining approach," in *Neural Information Processing*, (Lecture Notes in Computer Science vol. 5864). New York: Springer-Verlag, 2009, pp. 777–785.

[110] D. Dasgupta, Z. Michalewicz, and D. DasGupta, *Evolutionary Algorithms in Engineering Applications*. New York: Springer-Verlag, 1997.

[111] S. Todd and W. Latham, *Evolutionary Art and Computers*. New York: Academic, 1994.

[112] A. Kernytsky and B. Rost, "Using genetic algorithms to select most predictive protein features," *Proteins: Struct., Funct., Bioinformat.*, vol. 75, no. 1, pp. 75–88, 2009.

[113] J. Du, L. Xie, and S. Schroeder, "Pin optimal distribution of auction vehicles system: Applying price forecasting, elasticity estimation, and genetic algorithms to used-vehicle distribution," *Market. Sci.*, vol. 28, no. 4, pp. 637–644, 2009.

[114] J. L. Chen and W.-D. Chang, "Feedback linearization control of a two-link robot using a multi-crossover genetic algorithm," *Expert Syst. Appl.*, vol. 36, no. 2, Part 2, pp. 4154–4159, 2009.

[115] U. Maulik, "Medical image segmentation using genetic algorithms," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 2, pp. 166–173, Mar. 2009.

[116] S.-F. Hwang and R.-S. He, "A hybrid real-parameter genetic algorithm for function optimization," *Adv. Eng. Informat.*, vol. 20, no. 1, pp. 7–21, 2006.

[117] U. Sanpakdee, A. Walairacht, and S. Walairacht, "Adaptive spam mail filtering using genetic algorithm," in *Proc. 8th Int. Adv. Commun. Technol. Conf.*, 2006, vol. 1, pp. 441–445.

[118] J. Dudley, L. Barone, and L. While, "Multi-objective spam filtering using an evolutionary algorithm," in *Proc. IEEE Congr. Evol. Comput.*, 2008, pp. 123–130.

[119] M. S. Abadeh, J. Habibi, and C. Lucas, "Intrusion detection using a fuzzy genetics-based learning algorithm," *J. Netw. Comput. Appl.*, vol. 30, no. 1, pp. 414–428, 2007.

[120] S. Bridges and R. Vaughn, "Fuzzy data mining and genetic algorithms applied to intrusion detection," in *Proc. 23rd Nat. Inf. Syst. Security Conf.*, 2000, pp. 16–19.

[121] R. H. Gong, M. Zulkernine, and P. Abolmaesumi, "A software implementation of a genetic algorithm based approach to network intrusion detection," in *Proc. 6th Int. Conf. Softw. Eng., Artif. Intell., Network. Parallel/Distrib. Comput. and 1st ACIS Int. Workshop Self-Assembl. Wireless Netw.*, 2005, pp. 246–253.

[122] J. Gomez and D. Dasgupta, "Evolving fuzzy classifiers for intrusion detection," in *Proc. 2002 IEEE Workshop Inf. Assur.*, 2002, vol. 6, pp. 321–323.

[123] P. Espejo, S. Ventura, and F. Herrera, "A survey on the application of genetic programming to classification," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 2, pp. 121–144, Mar. 2010.

[124] M. Crosbie and G. Spafford, "Applying genetic programming to intrusion detection," in *Proc. Work. Notes AAAI Symp. Genet. Programm.*, 1995, pp. 1–8.

[125] A. Orfila, J. M. Estevez-Tapiador, and A. Ribagorda, "Evolving high-speed, easy-to-understand network intrusion detection rules with genetic programming," in *Applications of Evolutionary Computing*, (Lecture

Notes in Computer Science vol. 5484). New York: Springer-Verlag, 2009, pp. 93–98.
[126] G. Suarez-Tangil, E. Palomar, J. de Fuentes, J. Blasco, and A. Ribagorda, "Automatic rule generation based on genetic programming for event correlation," in *Computational Intelligence in Security for Information Systems*, (Advances in Intelligent and Soft Computing vol. 63). New York: Springer-Verlag, 2009, pp. 127–134.
[127] S. Mabu, C. Chen, N. Lu, K. Shimada, and K. Hirasawa, "An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 1, pp. 130–139, Jan. 2011.
[128] Y. Feng, J. Zhong, Z.-y. Xiong, C.-x. Ye, and K.-g. Wu, "Network anomaly detection based on DSOM and ACO clustering," in *Advances in Neural Networks–ISNN 2007*, D. Liu, S. Fei, Z. Hou, H. Zhang, and C. Sun, Eds., (Lecture Notes in Computer Science vol. 4492) New York: Springer-Verlag, 2007, pp. 947–955.
[129] C.-H. Tsang and S. Kwong, "Ant colony clustering and feature extraction for anomaly intrusion detection," in *Swarm Intelligence in Data Mining*, A. Abraham, C. Grosan, and V. Ramos, Eds., (Studies in Computational Intelligence vol. 34) New York: Springer-Verlag, 2006, pp. 101–123.
[130] E.-S. M. El-Alfy, "Discovering classification rules for email spam filtering with an ant colony optimization algorithm," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 1778–1783.
[131] H.-H. Gao, H.-H. Yang, and X.-Y. Wang, "Ant colony optimization based network intrusion feature selection and detection," in *Proc. Int. Mach. Learn. Cybern. Conf.*, Aug. 2005, vol. 6, pp. 3871–3875.
[132] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, 2nd ed. ed. New York: Wiley, 2007.
[133] L. Xiao, Z. Shao, and G. Liu, "K-means algorithm based on particle swarm optimization algorithm for anomaly intrusion detection," in *Proc. 6th World Congr. Intell. Control Autom.*, 2006, vol. 2, pp. 5854–5858.
[134] S. Srinoy, "Intrusion detection model based on particle swarm optimization and support vector machine," in *Proc. IEEE Symp. Comput. Intell. Security Defense Appl.*, Apr. 2007, pp. 186–192.
[135] C.-C. Lai and C.-H. Wu, "Particle swarm optimization-aided feature selection for spam email classification," in *Proc. 2nd Int. Innovat. Comput., Inf. Control Conf.*, Sep. 2007, pp. 165–168.
[136] L. N. de Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. New York: Springer-Verlag, 2002.
[137] J. E. Hunt and D. E. Cooke, "Learning using an artificial immune system," *J. Netw. Comput. Appl.*, vol. 19, no. 2, pp. 189–212, 1996.
[138] X. Yue, A. Abraham, Z. Chi, Y. Hao, and H. Mo, "Artificial immune system inspired behavior-based anti-spam filter," *Soft Comput.- Fusion Found., Methodol. Appl.*, vol. 11, no. 8, pp. 729–740, 2007.
[139] S. Sarafijanovic and J. Le Boudec, "Artificial immune system for collaborative spam filtering," *Stud. Comput. Intell.*, vol. 129, pp. 39–51, 2008.
[140] J. Kim, P. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, and J. Twycross, "Immune system approaches to intrusion detection—A review," *Nat. Comput.*, vol. 6, no. 4, pp. 413–466, 2007.
[141] C.-M. Ou and C. Ou, "Multi-agent artificial immune systems (MAAIS) for intrusion detection: Abstraction from danger theory," in *Agent and Multi-Agent Systems.: Technologies and Applications*, (Lecture Notes in Computer Science vol. 5559). New York: Springer-Verlag, 2009, pp. 11–19.
[142] A. Visconti, N. Fusi, and H. Tahayori, "Intrusion detection via artificial immune system: A performance-based approach," in *Biologically-Inspired Collaborative Computin* (IFIP International Federation for Information Processing). New York: Springer, 2008, pp. 125–135.
[143] T. Oda and T. White, "Developing an immunity to spam," in *Proc. Genetic Evol. Comput. Conf.*, 2003, pp. 231–242.
[144] T. Oda and T. White, "Increasing the accuracy of a spam-detecting artificial immune system," in *Proc. Congr. Evol. Comput.*, 2003, pp. 390–396.
[145] T. Oda and T. White, "Immunity from spam: An analysis of an artificial immune system," in *Proc. 4th Int. Artif. Immune Syst. Conf.*, 2005, pp. 276–289.
[146] A. Poteralski and M. Szczepanik, "Artificial immune optimization: Tests and comparison with evolutionary algorithm," in *Proc. 8th World Congr. Comput. Mech. 5th Eur. Congr. Comput. Methods Appl. Sci. Eng.*, 2008, pp. 1–2.
[147] C. R. Haag, G. B. Lamont, P. D. Williams, and G. L. Peterson, "An artificial immune system-inspired multiobjective evolutionary algorithm with application to the detection of distributed computer network intrusions," in *Proc. 2007 GECCO Conf. Compan. Genet. Evol. Comput*, 2007, pp. 2717–2724.
[148] Z. Qing-hua, Y.-z. Fu, and B.-g. Xu, "A new model of self-adaptive network intrusion detection," in *Proc. IEEE Congr. Evol. Comput.*, 2008, pp. 436–439.
[149] G. Prashanth, V. Prashanth, P. Jayashree, and N. Srinivasan, "Using random forests for network-based anomaly detection at active routers," in *Proc. Conf. Signal Process., Commun. Network.*, 2008, pp. 93–96.
[150] J. H. Holland, "Adaptation," *Progr. Theor. Biol.*, vol. 4, pp. 263–293, 1976.
[151] L. Bull, *Applications of Learning Classifier Systems*. New York: Springer-Verlag, 2004.
[152] J. Bacardit, B.-M. Ester, and M. V. Butz, "Learning classifier systems: Looking back and glimpsing ahead," in *Proc. 10th Int. Workshop Learn. Classifier Syst.*, 2008, pp. 1–21.
[153] K. Tamee, P. Rojanavasu, S. Udomthanapong, and O. Pinngern, "Using self-organizing maps with learning classifier system for intrusion detection," in *PRICAI 2008: Trends in Artificial Intelligence*, (Lecture Notes in Computer Science). New York: Springer-Verlag, 2008, pp. 1071–1076.
[154] M. Behdad, L. Barone, T. French, and M. Bennamoun, "On XCSR for electronic fraud detection," *Evol. Intell.*, vol. 5, pp. 139–150, 2012.
[155] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS Int. Trans. Comput. Sci. Eng.*, vol. 30, no. 1, pp. 25–36, 2006.
[156] P. Yang, L. Xu, B. Zhou, Z. Zhang, and A. Zomaya, "A particle swarm based hybrid system for imbalanced medical data sampling," *BMC Genom.*, vol. 10, no. Suppl 3, 2009, pp. 1–14.
[157] S. García and F. Herrera, "Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy," *Evol. Comput.*, vol. 17, no. 3, pp. 275–306, Sep. 2009.
[158] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special issue on learning from imbalanced data sets," *ACM SIGKDD Explorat. Newslett.*, vol. 6, no. 1, pp. 1–6, 2004.
[159] G. Giacinto, R. Perdisci, M. D. Rio, and F. Roli, "Intrusion detection in computer networks by a modular ensemble of one-class classifiers," *Inf. Fus.*, vol. 9, no. 1, pp. 69–82, 2008.
[160] N. Japkowicz, "Supervised versus unsupervised binary-learning by feedforward neural networks," *Mach. Learn.*, vol. 42, pp. 97–122, 2001.
[161] U. Bhowan, M. Johnston, and M. Zhang, "Developing new fitness functions in genetic programming for classification with unbalanced data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 406–421, Apr. 2012.
[162] A. Orriols and E. Bernado-Mansilla, "Class imbalance problem in UCS classifier system: Fitness adaptation," in *Proc. IEEE Congr. Evol. Comput.*, 2005, vol. 1, pp. 604–611.
[163] A. Orriols-Puig and E. Bernadó-Mansilla, "Evolutionary rule-based systems for imbalanced data sets," *Soft Comput.—A Fus. Found., Methodol. Appl.*, vol. 13, no. 3, pp. 213–225, 2009.
[164] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial classification," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2004, pp. 99–108.
[165] A. Patcha and J.-M. Park, "A game theoretic approach to modeling intrusion detection in mobile ad hoc networks," in *Proc. Inf. Assur. Workshop, 2004. Proc. 5th Annu. IEEE SMC*, Jun. 2004, pp. 280–284.
[166] I. Koychev, "Gradual forgetting for adaptation to concept drift," in *Proc. ECAI 2000 Workshop Curr. Issues Spatio-Temporal Reason.*, 2000, pp. 101–106.
[167] M. M. Lazarescu, S. Venkatesh, and H. H. Bui, "Using multiple windows to track concept drift," *Intell. Data Anal.*, vol. 8, no. 1, pp. 29–59, 2004.
[168] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: An ensemble method for drifting concepts," *J. Mach. Learn. Res.*, vol. 8, pp. 2755–2790, 2007.
[169] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. San Mateo, CA: Morgan Kaufmann, 2006.
[170] H. Beyer, M. Olhofer, and B. Sendhoff, "On the impact of systematic noise on the evolutionary optimization performance—A sphere model analysis," *Genet. Programm. Evol. Mach.*, vol. 5, no. 4, pp. 327–360, 2004.
[171] D. Gamberger, N. Lavrac, and S. Dzeroski, "Noise detection and elimination in data preprocessing: Experiments in medical domains," *Appl. Artif. Intell.*, vol. 14, no. 2, pp. 205–223, 2000.
[172] A. Di Pietro, L. While, and L. Barone, "Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions," in *Proc. Congr. Evol. Comput.*, 2004, vol. 2, pp. 1254–1261.
[173] S. Prestwich, S. Tarim, R. Rossi, and B. Hnich, "A steady-state genetic algorithm with resampling for noisy inventory control," in *Proc. 10th Int. Conf. Parallel Probl. Solv. Nature*, 2008, pp. 559–568.

[174] T. A. Almeida, A. Yamakami, and J. Almeida, "Filtering spams using the minimum description length principle," in *Proc. 2010 ACM Symp. Appl. Comput.*, 2010, pp. 1854–1858.
[175] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognit.*, vol. 40, no. 12, pp. 3358–3378, 2007.
[176] A. Voss and J. Voss, "A fast numerical algorithm for the estimation of diffusion model parameters," *J. Math. Psychol.*, vol. 52, no. 1, pp. 1–9, 2008.
[177] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
[178] M. A. Franco, N. Krasnogor, and J. Bacardit, "Speeding up the evaluation of evolutionary learning systems using GPGPUs," in *Proc. 12th Annu. Genet. Evol. Comput. Conf.*, 2010, pp. 1039–1046.
[179] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Trans. Automat. Control*, vol. AC-26, no. 1, pp. 17–32, Feb. 1981.
[180] M. Behdad, T. French, L. Barone, and M. Bennamoun, "On principal component analysis for high-dimensional XCSR," *Evol. Intell.*, vol. 5, pp. 129–138, 2012.
[181] J. Bacardit and N. Krasnogor, "Fast rule representation for continuous attributes in genetics-based machine learning," in *Proc. 10th Annu. Genet. Evol. Comput. Conf.*, 2008, pp. 1421–1422.
[182] B. Gavish and C. L. Tucci, "Reducing internet auction fraud," *Commun. ACM*, vol. 51, no. 5, pp. 89–97, 2008.
[183] K. C. Wilbur and Y. Zhu, "Click fraud," *Market. Sci.*, vol. 28, no. 2, pp. 293–308, 2009.
[184] Microsoft News Center. (2010). "Microsoft investigators uncover emerging form of click fraud," [Online]. Available:https://www.microsoft.com/Presspass/press/2010/may10/05-19ClickFraudPR.mspx
[185] T. Moore and R. Clayton, "The impact of incentives on notice and takedown," in *Managing Information Risk and the Economics of Security*. New York: Springer-Verlag, 2009, pp. 199–223.
[186] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi, "Detecting spam blogs: A machine learning approach," in *Proc. 21st Nat. Conf. Artif. Intell.*, 2006, pp. 1351–1356.
[187] N. Jindal and B. Liu, "Analyzing and detecting review spam," in *Proc. 7th IEEE Int. Conf. Data Mining*, 2007, pp. 547–552.
[188] N. Jindal and B. Liu, "Review spam detection," in *Proc. 16th Int. Conf. World Wide Web.*, 2007, pp. 1189–1190.
[189] M. Potthast, B. Stein, and R. Gerling, "Automatic vandalism detection in wikipedia," in *Advances in Information Retrieval*, C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, and R. White, Eds., (Lecture Notes in Computer Science vol. 4956) Berlin/Heidelberg, Germany: Springer-Verlag, 2008, pp. 663–668.
[190] R. Lukashenko, V. Graudina, and J. Grundspenkis, "Computer-based plagiarism detection methods and tools: An overview," in *Proc. 2007 Int. Comput. Syst. Technol. Conf.*. New York: ACM, 2007, pp. 1–6.
[191] P. Kolari, T. Finin, and A. Joshi, "SVMs for the blogosphere: Blog identification and splog detection," in *Proc. AAAI Spring Symp. Comput. Approaches Anal. Weblogs*, 2006, pp. 92–99.

Mohammad Behdad received the M.Sc. degree. He is currently working toward the Ph.D. degree the School of Computer Science and Software Engineering, University of Western Australia, W.A., Australia.

He has been a Lecturer in the subjects of software engineering, computer networks, and database systems. His research interests include genetics-based machine-learning systems and their applications in the electronic fraud detection systems.

Luigi Barone received his Ph.D. degree from The University of Western Australia where he is currently a lecturer.

He has more than 15 years of research experience in the field of computational intelligence, investigating both the theoretical aspects and practical applications of such approaches. His main contributions have been to the field of multiobjective evolutionary algorithms, learning in computer games, and evolutionary design. He has published more than 50 scientific articles and won three competitive research grants.

Mohammed Bennamoun received the M.Sc. degree control theory and the Ph.D. degree in the area of computer vision.

He is currently a Winthrop Professor with the University of Western Australia, W.A., Australia. He is the coauthor of two books. He published more than 170 journals and conference publications, and secured highly competitive national grants from the Australian Research Council. His areas of interests include control theory, robotics, obstacle avoidance, object recognition, artificial neural networks, signal/image processing, and computer vision (particularly 3-D).

Tim French received his Ph.D. degree from The University of Western Australia (UWA), W.A., Australia.

He is currently a Lecturer in the School of Computer Science and Software Engineering, University of Western Australia, W.A., Australia. He works in the area of logics for multiagent systems and formal methods for software engineering. He has written several papers on the decidability and expressivity of extensions of modal logics and has more recently been working on the application of these ideas to awareness and certainty in multiagent systems.