Journal of Network and Computer Applications 72 (2016) 14–27

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

User profiling in intrusion detection: A review

Jian Peng [a], Kim-Kwang Raymond Choo [b,a,c,*], Helen Ashman [a]

[a] School of Information Technology and Mathematical Sciences, University of South Australia, Australia
[b] Department of Information Systems and Cyber Security, University of Texas at San Antonio, USA
[c] School of Computer Science, China University of Geosciences, Wuhan, China

ARTICLE INFO

Article history:
Received 23 February 2016
Received in revised form
20 June 2016
Accepted 29 June 2016
Available online 1 July 2016

Keywords:
Behavioural biometrics
Intrusion detection and prevention systems
Psychometrics
User behaviour
User profiling

ABSTRACT

Intrusion detection systems are important for detecting and reacting to the presence of unauthorised users of a network or system. They observe the actions of the system and its users and make decisions about the legitimacy of the activity and users. Much work on intrusion detection has focused on analysing the actions triggered by users, determining that atypical or disallowed actions may represent unauthorised use. It is also feasible to observe the users' own behaviour to see if they are acting in their'usual' way, reporting on any sufficiently-aberrant behaviour. Doing this requires a user profile, a feature found more often in marketing and education, but increasingly in security contexts. In this paper, we survey literature on intrusion detection and prevention systems from the viewpoint of exploiting the behaviour of the user in the context of their user profile to confirm or deny the legitimacy of their presence on the system (i.e. review of intrusion detection and prevention systems aimed at user profiling). User behaviour can be measured with both behavioural biometrics, such as keystroke speeds or mouse use, but also psychometrics which measure higher-order cognitive functions such as language and preferences.

Contents

1. Introduction

An intrusion detection system (IDS) monitors host systems and/or network traffic for suspicious activity. Once it finds any, it alerts the system or network administrator. In some cases, the IDS may also respond to anomalous or malicious traffic by taking

* Corresponding author at: Department of Information Systems and Cyber Security, University of Texas at San Antonio, USA.
E-mail address: raymond.choo@fulbrightmail.org (K.-K. Choo).

http://dx.doi.org/10.1016/j.jnca.2016.06.012
1084-8045/

action such as blocking the user or source IP address from accessing the network.

Intrusion detection systems are generally classified according to where they perform their observations. An IDS can be *network-based* or *host-based*. A network-based IDS observes strategic points within the network to monitor traffic to and from all devices on the network. In contrast, a host-based IDS runs on an individual host or device on the network, monitors the inbound and outbound packets from that device only and alerts the user or administrator if suspicious activity is detected.

Besides these two types of IDS, another proposed by Pennington et al. (2010) is storage-*based* intrusion detection, which analyses all requests received by the storage server and determines the system intrusions by the profiles of data access patterns of systems. As there are lots of logs/traces on storage devices, they can be used for intrusion analysis (Khan et al., 2016). The advantages are that it can be independent from the client's operating systems and continues to identify the intrusions after systems have been compromised, whereas host-based and network-based IDS can comparably be easier be disabled by the intruder; since storage devices are often on different platforms, having restricted interface to outside, it can be more difficult for intruders to compromise them and delete their attack logs and traces which have also been used in forensic investigations This type of IDS are generally used for intrusion detection in storage area network, object-based storage devices, workstation disk drives (Rahman and Choo, 2015; Martini and Choo 2014; Quick et al., 2013; Yampolskiy and Govindaraju, 2008).

IDSs are also often classified according to their primary technique, and can be either *signature-based* (also known as rule-based), or *anomaly-based*. The signature-based IDS monitors packets on the network and compares them against a database of signatures or attributes from known, previously-established malicious threats. This is similar to the way most antivirus software detects malware (Rhodes et al., 2000; Alexandre, 1997; Cortes and Pregibon, 2001; Han et al., 2002; Venugopala and Hu, 2008; Blasing et al., 2010). Although this technique is considered the *de facto* standard, a key limitation is the delay associated with updating the IDS signatures of new intrusions(Afroz et al., 2012), and during that time the IDS is unable to detect the new threat (e.g. zero-day vulnerabilities).

In contrast, the anomaly-based IDS technique is able to detect new forms of attack without prior notification of them. Instead it monitors network traffic and compares it against an established baseline, where the baseline identifies what is “normal” for that network, what protocols are generally used, what ports and devices generally connect to each other. It alerts the administrator or user when anomalous or significantly different traffic is detected (Keselj et al., 2003; Barron-Cedeno et al., 2010; Marceau, 2000; Shrestha and Solorio, 2013; Houvardas and Stamatatos, 2006). However, it may miss both known and novel attacks if they are not manifested along the observed dimensions. Also, depending on how finely-tuned the analysis is, it can have a high error rate, either alerting genuine behaviour as an intrusion (i.e. a false positive) or conversely, not detecting an intruder (i.e. a false negative). Additionally, it needs purity of training data, i.e. an absence of attacks when creating the initial baseline against which to compare later activity. Finally, it is a *post facto* technique which can only detect an attack once it has already occurred, and which may be easy to evade once the model is known.

A typical IDS (Denning, 1987; Mitchell and Chen, 2014; Yeung and Ding, 2002) includes the following components:

1) Data collector collects relevant data from the sensors on monitored devices or systems.
2) Profile generator analyse the data from the Data collector and generate profiles. The anomaly based IDS builds the normal profiles automatically but a signature based IDS may involve experts' efforts to generate its malicious signatures during the training stage.
3) Profile templates store the profiles and are shaded in Fig. 1. A signature based IDS saves malicious profiles (signatures) while an anomaly based IDS saves normal profiles.
4) Intrusion detector is the key component in the IDS and carries out the task of detecting intrusion based on the current profiles.
5) Post processor is responsible proper actions taken once intrusion take place.
6) Profile updater makes proper updates based on current received profiles and relevant algorithms (Fig. 1).

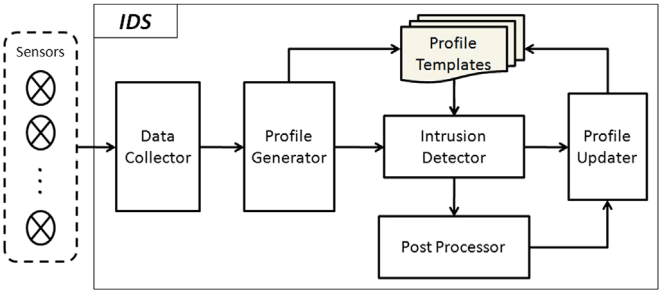

Fig. 1. A typical IDS architecture.

Clearly, there is benefit in operating complementary approaches to intrusion detection, not just in signature-based and anomaly detection techniques, but in a combination of anomaly detection profiles. Keystroke analysis is highly effective for intrusion detection but will throw up false positives if, for example, the user has an arm injury which causes them to type differently. However, combining it with other profiles, such as habitual web sites, favoured applications, normal access time, and so on, will help ameliorate detection errors generated by singular deviations in one profile.

Behavioural science is concerned with gaining a better understanding of human behaviour which focuses specifically on criminal human behaviour in an attempt to better understand criminals—who they are, how they think, why they do what they do—as a means to help solve malicious intrusions. A definition of “behavioural profiling” as offender profiling suggests techniques used to identify likely suspects and analyse patterns that may predict future offences and/or victims (Woodhams, Toye). These techniques are able to help investigators to accurately predict and profile the profiles of unknown criminal subjects or offenders. Behavioural profiling can be either used to identify a potential intruder or to determine normal user patterns, but it is much harder to profile an anomaly behaviour as intruders are often intentionally employ some measures for evasion (Maor, 2013).

Unlike the surveys discussed above, Abdel-Hafez and Xu (2013) discuss and compare existing user modelling techniques for social media sites. They also explain how user profiles are constructed in their modelling process. Jin et al. (2013) review user behaviours in online social networks by social connectivity, interaction among users, and traffic activity. They also analyse malicious behaviours of online social network users and proposed solutions to detect misbehaving users. The focus is, however, on user social behaviours rather than security. Stamatatos (2009) surveys automated approaches to attributing authorship by examining their profiles for both text representation and text classification. However, the focus of this survey is on computational requirements and settings rather than on linguistic or literary issues. Rodríguez et al. (2014) classify human activity recognition methods as data-driven and knowledge-based techniques and use them to represent human

activities. They use ambient intelligence extract user behaviours such as daily life which are far different from ours.

Only a small number of existing surveys briefly discuss user profiling, and to our best knowledge, there is no existing survey of user profiling for intrusion detection. This is the gap we seek to address in this paper. Specifically, we focus on user profiling by first analysing behaviours and categorising them into system behaviours and user behaviours. Based on this classification, profiles are grouped into system profiles and user profiles accordingly, with the latter being further ramified into more specific categories. These specific profiles and related analysis techniques, and their merits and limitations, are then summarised. The publications included in this survey were located by searching on Google Scholar and other academic databases (i.e. ACM Digital Library, IEEE Xplore, ScienceDirect and Springer) using keywords such as "signature-based intrusion detection", "anomaly-based intrusion detection", "biometric-based intrusion detection","psychometric intrusion detection", and "behaviour profiling".

The rest of this paper is organized as follows. We review anomaly-based intrusion detection in Section 2, focusing on behaviour profiling in anomaly-based IDSs in Section 3. We then discuss user modelling and the profiles used in user models in Section 4 before focusing on biometric and psychometric user profiles in Section 5. Finally, the paper ends with the conclusions and future opportunities in Section 6.

2. Background

There are a number of published surveys on intrusion detection. In 2014, for example, Mitchell and Chen (2014a) study intrusion detection techniques for cyber-physical systems based on detection technique and audit material, and they summarise the advantages and drawbacks of existing approaches. In the same year, the authors also reviewed the intrusion detection literature for wireless networks (Mitchell and Chen, 2014a). A number of research gaps were identified. However, in both surveys (Mitchell and Chen, 2014b; Yeung and Ding, 2002), Mitchell and Chen focus on behaviour-based intrusion techniques.

Another similar survey on anomaly-based intrusion detection techniques for MANET was undertaken by Kheyri and Karami (2012), although the survey focused only on detection methods, rather than feature profiling and modelling (which is the focus of this paper). Patcha and Park (2007) propose a generic IDS architectural design, as well as investigating existing detection techniques used in anomaly detection, but user profiling is not included in their survey. Mohammad Faysel (2010) analyse published intrusion prevention the techniques, and Chandola et al. (2009) classify existing anomaly detection techniques into different categories based on the underlying approach and application domains. Based on the classification, Chandola et al. (2009) compare the computational complexity for the studied techniques, however, user profiling is not included. In a related survey by the same authors (Chandola and Kumar, 2012), they propose a classification for discrete sequences based on the problem formulation to identify distinct sequence. In addition, they discuss the relevance of their approach for various application domains. Jacob et al. (2008) survey reasoning techniques used by malware behavioural detectors and classify the detectors into different categories according to different criteria. In the analysis, they build average profiles for each class of malware in clustering analysis during training process. Monowar et al. (2013) survey existing network anomaly detection methods and systems based on the underlying computational techniques and preventing tools. They use network traffic profiles in their matching process and regard them as one type of reference data.

An anomaly-based IDS can handle new forms of attacks by establishing whether the observed behaviour is'normal' (i.e. a situation without attacks). As pointed out by one of the reviewers,'this is really difficult to obtain in practice, but this is a well-known problem in the anomaly based IDS field'. Whether the observed behaviour is classed as normal or not is determined by how disparate the data is from its normal value in a statistical model obtained by the audit log data. Its implementation includes two phases: one is the learning phase in which a'normal' model is built, and the other is the detection phase where current behaviours are matched against the normal model and a warning is triggered when it finds discrepancies.

Some models are static and others are dynamic. Static models do not take temporal factors into account and mainly refer to features of overall data while dynamic models regard timing as one of many variables during modelling. Yeung and Ding (2002) show that a dynamic model achieves better performance than the static one in most cases and that the frequency analysis of different shell commands generally produced better results. Dynamic models may be essential to analyse data with time of day unless a static model is applied to blocks of times. For example, a day may be split up into hours and analysis performed per hour and compared with similar hours of different days.

There is a variety of techniques implementing these decisions in anomaly-based intrusion detection systems, such as statistical approaches (Anderson, 1980; Umphress and Williams, 1985; Pannell and Ashman, 2010; Chen et al., 2013), artificial neural networks (Alexandre, 1997; Li et al., 2006; Vizer et al., 2009), hidden Markov models (Li et al., 2012), rule learning (Denning, 1987; Tabia and Benferhat, 2008), decision trees (Stein et al., 2005), support vector and machines (Afroz et al., 2012; Houvardas and Stamatatos, 2006; Masud et al., 2007). Each has its merits and drawbacks, for example, artificial neural networks are similar to the statistical approach to some extent, but artificial neural networks is easier to represent nonlinear relationships between inputs and outputs regardless of distortions and incompleteness of data. However, it is harder to interpret the relationships between these inputs and outputs. On the other hand, a sequence fuzzy-based approach is better at learning (Chebrolua et al., 2005). Stein et al. (2005) research both the decision tree and genetic algorithm for data classification and propose a hybrid system of both. Their results looked at classifying network data, not user behaviour, and show that the hybrid approach can improve classifying performance over both decision tree and genetic algorithm individually. Vizer's team employ multiple techniques such as artificial neural networks (Alexandre, 1997; Chebrolua et al., 2005; Gates and Taylor, 2006) for learning, decision trees (Yeung and Ding, 2002; Chebrolua et al., 2005) for classification, support vector machines (Chebrolua et al., 2005; Gates and Taylor, 2006; Kandias et al., 2010; Stein et al., 2010; Iqbal et al., 2010) for good performance and optimal hyper-planes in feature space, and k-nearest neighbours (Vizer et al., 2009) for processing keyboard behaviours.

However, Tabia and Benferhat (2008) notice that the tree size could reduce the number of possible decision rules and conjecture that the decision tree may not suitable for detecting new intrusions. They proposed to relax the minimum description length of the tree to increase the detection rates. The result shows that this relaxation can avoid creating large trees, thus improving the detection speed performance.

Both signature-based and anomaly-based detections have an obvious drawback in common: they require prior knowledge of some sort, in the form of attack signatures or normal activity profiles, in order to operate (Osanaiye et al., 2016a,b). Casas et al. (2012) introduced an unsupervised network IDS without prior knowledge by determining outliers based on sub-space clustering and multiple Evidence Accumulation approaches. They put traffic

anomalies into two categories based on IP flows and their spatial structures: 1-to-N anomalies (many IP flows from a single source to different destinations) and N-to-1 anomalies (many IP flows from many sources to a single destination). The 1-to-1 is regarded a specific case and N-to-N. In order to degrade the complexity of an N-to-N anomaly, it can be regarded as an N-to-1 anomaly with a 1-to-N anomaly. They define anomalies which are significantly different from the rest of their sample data. They also take advantage of the divide-and-conquer technique to overcome the drawback of the lack of robustness by the classifier algorithms. One disadvantage of their approach is that it runs slowly. The other one is of the high false positive rate (Casas et al., 2012).

There is a wide range of profiles or behaviours that anomaly detection analyses can be performed over. We look at these in the next section.

3. Behavioural-based IDS

Behavioural intrusion detection systems focus on behaviours. The behaviours can be divided into *system behaviours* and *user behaviours*. The system behaviours are generated by hosts and networks and relate to the host activities and network status; this is further discussed in Section 3.1. In contrast, the user behaviours mainly relate to the direct interaction between the user and the system, for example, typing patterns; this is further discussed in Section 3.2. However, these two types of behaviour are loosely related and one can affect the other. For example, if a user is discussing some topics online, the activities of the CPU, memory and network generated by the user's actions can be regarded as system behaviours, while the profiles associated with their typing and text are user behaviours (Fig. 2).

3.1. System behaviours

Most existing studies are based on system behaviours, in part because data can be obtained relatively more easily and without privacy issues (Denning, 1987; Yeung and Ding, 2002; Umphress and Williams, 1985; Forrest et al., 1996; Bergadano et al., 2003; Maxion, 2003).

Forrest et al. (1996) focus on profiling processes in a system via analysing the execution paths of system calls (e.g. “sendmail” in Unix) sent by the process to the operating system. Their results showed that their method could find about 75% of anomalous system calls. Marceau (2000) and Cabrera et al. (2001) undertook similar experiments using different methods. The former uses multiple length n-grams on system logs and shows a high rate (13/20) of finding anomalies with fewer datasets than Forrest and Longstaff, and by analysing the normal dictionary, which stores the permitted sequences of system calls, shows that it was possible to detect anomalies with a detection rate of 75–100%.

Mazzariello and Oliviero (2006) look at a system to profile users where a network intrusion detection system is placed on a network, and traffic from hosts is sniffed and analysed. The traffic generated is used to create a user profile by looking at different features such as protocols, connection status, destination, source, time sent, and frequency. This work could be further extended to place a system on the host to analyse traffic instead from a remote location; however, using network traffic to profile a user may contain a lot of “noise” as applications may send traffic that is not necessarily related to user behaviour e.g. automatic update checking and keep-alive pings. As they only propose a detection approach and a framework but no implementation, no performance assessments are available.

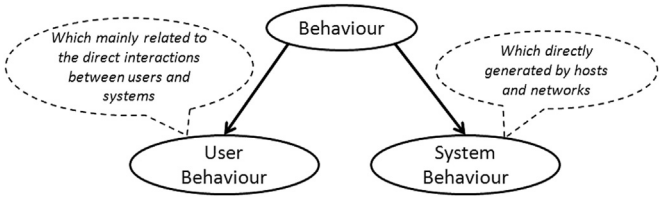

Fig. 2. Classification of behaviours.

Other anomaly-based detection methods proposed include the workflow-based method (Li et al., 2012) which uses a series of HTTP sessions to detect anomalous behaviour of users. The sessions are first broken down into data object workflows, and then a hidden Markov model is created based on this workflow. Their results show their approach achieves a high detection rate of 96.93% and false positive rate of 2.94%. The drawbacks include less flexibility for workflow variances and lack of consideration of user behaviours.

Although techniques based on system behaviours have been very successful, they inherit some drawbacks: they can be falsified and may require an additional device to capture them (Alexandre, 1997). To complement system behaviour methods, approaches based on user behaviours are used for intrusion detection.

3.2. User behaviours

Strictly speaking, all system behaviours are generated, directly or indirectly, by human users. Different interactions between a user's actions and a system produce distinct system behaviours. A frequent MS-Word user can generate a high level of disk I/O operations, while a Web user may introduce additional network traffic.

Vizer et al. (2009) take a slightly different approach to user profiling and keystroke analysis by applying the differentiation of key delays and typing patterns to determine whether the user is cognitively or physically stressed. While this research is not directly relevant to intrusion detection, it shows that anomalistic behaviour in some user profiles could help diagnose health problems of the user.

A system by Li et al. (2006) tried to find anomalous user activity against user behaviour models and specific constraints by analysing Microsoft Windows NT and audit data such as process CPU usage, applications running and number of windows open. They used neural networks, and support vector machines, finding that the neural network was able to achieve a significantly higher detection rate of 90% when compared to the support vector machine rate of 63%. However, the support vector machine algorithm was able to provide a lower false-positive rate – 3.7% vs. 10% for the neural network. They declare that their system requires much less training data with only the user's own legitimate data.

Alexandre (1997) employs a biometric method combined with a password to recognise behaviours of keyboard signatures, which are expressed by a sequence of keystrokes. Their assumption is that user behaviour is more difficult to falsify compared to the mechanism of authentication and access control by passwords, because anyone can log onto a system if the input credential is eligible, but they will be rejected even if the correct password is input if the keystroke dynamics are different. The artificial neural network and self-organizing techniques are adopted to analyse these signatures. The result shows that the method can achieve false acceptance rate of 0.092%.

Authorship attribution methods can also identify text according to its match with patterns or types. One use for this is in malware detection. As malware has different types and behaviours, the same malware detection methods and mechanisms cannot be used in all cases. An advanced one is called the suspicious behaviour approach which monitors the behaviours of all programs

(Rhee et al., 2011; Xie et al., 2010).

We will now look more closely at user modelling in general and how intrusion detection may benefit from it.

4. Profiles in IDS modelling

An anomaly-based IDS requires a user model that profiles the normal behaviour for that user so as to enable identification of anomalies in comparison to the normal behaviour of the user. The quality of the model, specifically the selected profiles and the relative importance given to each, is a key factor in the IDS. Automatic user modelling is not trivial, particularly when needing to deal with ill-balanced data distribution or very large quantities of data, deciding what the proper threshold is between the'normal' and the'abnormal', and adapting to constantly-changing environments (Wu and Banzhaf, 2010).

The objective of user modelling is to explicitly represent the properties of an individual user including needs and preferences as well as physical, cognitive and behavioural profiles. It is widely-used in many areas, such as recommender systems, and hypermedia education systems, as well as in marketing applications. While this may initially appear unrelated to intrusion detection, non-IDS applications tend to create a profile for the purpose of finding more items that somehow complement the same as those in the profile already, while IDS use of user profiles is more interested in when apparent users do not fit the model (Rodríguez et al., 2014). In IDS applications, user profiling can model legitimate users' behaviours for anomaly-based intrusion detection either by creating a model of each individual user, or by defining "typical" users, regardless of whether those typical users are representative of a larger group of users with "similar" interests or skills as in recommendation systems or the group of permitted users as in intrusion detection systems.

Behaviour-based user modelling has been present in the academic literature for some time, as well as being present in real-world applications such as Amazon's recommendation system since the 1990s. In adaptive hypermedia systems, Brusilovsky (2001) presents and classifies a list of user-modelling systems, known as "adaptive" systems, according to such things as whether the data about a user is collected explicitly (such as in the user's specified preferences) or implicitly (as a by-product of their interaction with a system). He also explains some valuable key concepts such as user profiles and response of the system to variations in user profiles which can apply equally to intrusion detection systems to detect anomalies in user behaviour. He also mentions the environment as including the user's geographic location which can be relevant to personalisation systems that perform localisation (for example, translation), and these too can be used in intrusion detection. He suggests that it may not be possible to comprehensively model an individual user simply by their preferences and that it may not be possible to predict preferences based on their individual trains of activity. The difficulty in accurately modelling the user is less relevant in behavioural IDS as the system is not trying to double-guess what the user needs or requires, but rather is trying to ensure that the apparent user and the profiled user are the same person. Thus behavioural IDS focus on profiles that can rapidly identify the user, and do not require an interpretation of their preferences in order operate well. Conversely, behavioural profiles, though showing some promise as identifiers or otherwise of individuals when used in intrusion detection systems, are not used in adaptive hypermedia, personalisation or recommender systems, as the operation of these latter systems requires an interpretation of preferences, skills and other high-level cognitive functions.

Another difference is the focus on psychometric data in traditional user modelling applications, as opposed to the collection of behavioural biometric data that is prevalent in behavioural intrusion detection systems. This is generally because the aim of non-IDS user modelling systems is to comprehend the user's preferences and personal traits in order to recommend them other items that may appeal to them, or to select further learning materials improve their understanding in educational systems. In contrast, behavioural intrusion detection systems have not yet made significant use of user preferences, relying primarily on measuring the user's motor skills or other similar behavioural biometric data.

An interesting finding is that more personal biometric profiles can be more efficient for anomaly detection, such as keystroke usage being more promising than CPU usage, memory usage and window usage. Experiments show that an anomalous user can be detected in 90 s (Pannell and Ashman, 2010).

However, psychometric data could be similarly useful for intrusion detection purposes. For example, a user's UNIX skills may be poor while an intruder's may be high, and a mismatch could accurately indicate an intrusion. Likewise preference for applications could indicate mismatches, especially where these are associated with different skill levels (e.g. using "vi" as opposed to "textedit"). It is feasible that a combination of both biometric and psychometric data could help improve detection performance although there are few instances of this to date.

As previously stated that behaviours can be categorised system behaviours and use behaviours, there are two types of profiling methods, namely: *system profiling* and *user profiling*. System profiling describes system behaviours while user profiling depicts users' behaviours (Fig. 3).

4.1. System profiles

System profiles can be further categorised into *host-related profiles* and *network-related profiles*. The former includes profiles produced on hosts such as file usage, CPU usage, I/O usage, program-related, GUI-related, and so on. The network-related profiles are those taken place on networks, such as sessions, and protocol-related (Fig. 3).

There have been a number of published works on intrusion detection via system profiling. For example, Eugene (2008) claimed that profiling user behaviour for intrusion detection originated from Anderson (1980), who introduces the idea of using an automated surveillance system that looks for profiles such as session logs, durations, program usage, device usage, and file usage. Anderson categorises users into three different user groups, namely: legitimate, masquerade, and clandestine. A legitimate user is an authorised user of the system, the masquerade is an intruder who attempts to use the system as a legitimate user, and a clandestine user, not necessarily pretending to be a legitimate user, is able to bypass the surveillance system altogether. The proposed surveillance system used basic statistics, such as

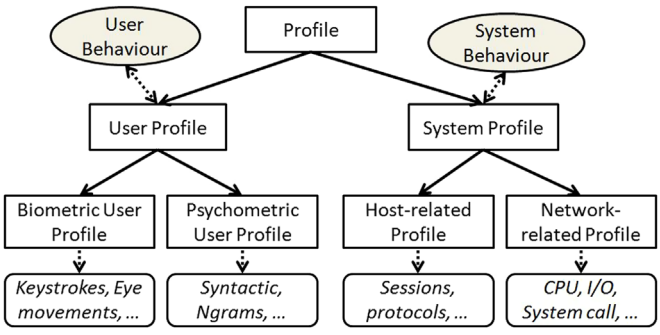

Fig. 3. Classifications of profiles.

averages, standard deviations, and maxima and minima to determine abnormal usage of the machine. However, this analysis method is not implemented.

Denning (1987) profiled users by basic statistics, such as user logins, CPU usage, and I/O usage, to determine an abnormality. Although there had not been an implemented system, many subsequent implementations of behavioural systems use some of the ideas presented by Denning. Based on Denning's work, Gates and Taylor (2006) found some flaws in some previous work and argued that better anomaly detection approaches, more reliable profiles and their combinations require further study. In the end, they suggest future research directions for implementing practical intrusion detection systems, such as the work by McKinney and Reeves (2009) who present a method of masquerade detection by a Naive Bayes classifier that analyse handle counts of processes as the input. The results show the detection rate can reach 97%.

4.2. User profiles

User profiling of individuals can be implemented in anomaly-based intrusion detection where there are two types of user profiling: *biometric* profiling and *psychometric* profiling. Biometric profiling deals with the user's behavioural biometric data, such as keystroke speed, and mouse and swipe screen use, that is, it profiles cognitively undemanding functions that the user often is not even aware of. In contrast, psychometric profiling mainly handles higher cognitive functions, such as writing, decision-making, choices and preferences. The main distinction between them is the amount of cognitive effort required, as most users do not consider their typing or mouse use, but will put much more thought into how they write, such as what words they choose (Fig. 3).

Note that psychometric profiling in the intrusion detection context refers to the statistical analysis of specific, observed data relating to higher-order cognitive functions, and is quite distinct from personality classification methods such as the Myers-Briggs Type Indicator (http://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/).

In practice, a profile of only a single characteristic of the user is not helpful in isolation for intrusion detection because of the possibility of false positives due to user incapacity or similar, and the combinations of different user profiles are often enabled in order to enhance detection performance.

In the next section, we will discuss biometric and psychometric user profiles.

5. Biometric and psychometric user profiles

We will now look more closely at biometric and psychometric user profiles, and how intrusion detection may benefit from these profiles.

5.1. Biometric user profiles

It is natural that different profiles require different input data to profile users. Thus, behavioural biometric data generally includes keystroke dynamics, swipe screen movements, and mouse dynamics, but could include less-common profiles such as eye movements, or even motion detection and facial features in systems with such capabilities, such as gaming consoles. The keystroke analysis involves gathering typing rhythms from the interaction between a user and computer, and is commonly known as keystroke dynamics (Fig. 3).

Umphress and Williams (1985) probably are the earliest to introduce the idea of keystrokes to verify users. Their authentication methods include the average time between all possible keystroke pairs and the average time between two specific keystrokes. Their results showed that the test profile matched well to the reference profile. This work is extended by Bergadano et al. (2003) via analysing both static and dynamic methods of keystroke data. The extracted features from the raw data include latency, duration and the order of the pressed keys. The results of their experiments show that their system can achieve more than 90% accuracy, for both authorizations and rejections. Pannell and Ashman (2010) also found that keystroke analysis enabled very rapid detection of false users, within about 90 s.

Revett (2009) proposes an anomaly detection method which uses the dynamics of the user typing profiles to describe users and achieve user authentication. It was assumed that each user's keystrokes were different and the features of the typing included key press durations, and multiple key latencies. The results showed that it was possible to achieve a 0.0% false-negative rate and 0.8% false-positive rate with enough enrolment (i.e. learning) data.

Bhaskaran et al. (2011) use eye movements to detect cheating behavioural profiles. They first use machine learning techniques to study underlying phenomena expressed when a person tells a lie, and then build a dynamic Bayesian model during training in normal conversation. The detection is based on measuring the deviation from normal behaviours. Their experiment shows that it can obtain an accuracy of 82.5% with 40 subjects. However, the usefulness of this approach is limited because it relies on the user's device to record data and either analyse or pass on for analysis the user's eye motion.

In fact, this need to trust the user's device to record and pass on user data is a significant limitation of all behavioural biometric profiling methods within an intrusion detection context. Trust is defined and well elaborated in the trust models by Wahab et al. (2015) and trust systems by Jøsang et al. (2007). Collecting data about a user's keystrokes, mouse and touchpad use or any other input device can only work when the user's device itself is trusted. However, intrusions do not necessarily (or even very often) originate from a trusted device. Indeed, the wise system security staff would assume that all remote devices not owned by the same company were at the least highly suspect. It is too easy to conjecture a scenario that involves an attacker sending false or modified keystroke data which has been adjusted to resemble that of a genuine user. For this reason, it becomes more valuable to consider the possibility of user-specific profiles that appear to be less susceptible to fraudulent modification. With this in mind, we turn to psychometric profiling.

5.2. Psychometric user profiles

Psychometric user profiles generally include data that reflect the user's intelligence, decisions, requirements, and preferences. These appear to be harder to falsify than biometric data, perhaps because each user's intelligence and tastes are significantly harder to deterministically model than such things as keystrokes which are easily reduced to purely numeric features without much loss of accuracy. The apparent difficulty that other, non-IDS user profiling applications have with double-guessing the user's needs or tastes is testament to the complexity of modelling user choices, personal taste and intelligence (Fig. 3). Numerous examples of poor recommendations due to inadequate user profiles abound, with ill-placed advertisements appearing in the press forming a regular topic for humour in one publication ("malgorithms" (www.private-eye.co.uk)).

User-written text includes rich source of highly personal data such as lexical, character, syntactic, semantic, and application-specific (Stamatatos, 2009; Chebrolua et al., 2005; Martin et al., 2005; Mezghani et al., 2012; Koppel et al., 2009; Davis and Clark,

2011; Peng et al., 2016). Generally, these profiles are obtained with higher costs as it is necessary to handle large quantities of data. However, as the technologies rapidly improve and the availability of electronic texts increases, it is feasible to adapt psychometric profiling for authorship attribution, plagiarism detection and malware detection. In particular authorship attribution is the key element for intrusion detection, as we wish to determine if the writer is who they say they are.

5.2.1. User profiles in authorship attribution

Authorship attribution can be dated back to the 19th century. Its main concern is to define an appropriate characterisation of documents that captures the writing style of the author. It often applies the knowledge and methods of linguistics to the context. It often provides information for the questions of authorship, such as "Who wrote the text?", "Do these words mean some other thing?" A famous example of early text analysis is related to the authenticity of Shakespeare's work (Frantzeskou et al., 2006) while the analysis of letters written by the "Unabomber" made it possible to attribute the letters to Theodore Kaczynski, enabling his prosecution. Recent work in authorship attribution demonstrates the practicality of automatically analysing documents based on authors' writing styles. During the last decade, this scientific field has developed substantially taking advantage of research advances in areas such as machine learning, information retrieval, and natural language processing.

Iqbal et al. (2008) used email 'write-prints' to determine authorship. The profiles they used for describing write-prints include morphology, syntax, and structure information in emails. Analysis approaches used are decision trees and support vector machines. They collected more than 200,000 email data from 158 employees of the Enron Company and filtered out high frequency words. Experimental results showed their method can achieve 80% accuracy when there are four suspects and 77% accuracy when there are 10 suspects. In follow-up research without training data (Iqbal et al., 2010), they extracted similar stylometric attributes for profiling authors. They used three clustering methods (Expectation Maximization, k-means clustering, and bisecting k-means clustering) to evaluate their method. Experimental results showed that the k-means method performed better when each cluster has fewer than 40 emails, while the bisecting k-means method was preferable when there is a large set of training data.

A similar approach to determine authorship of programs is introduced by Frantzeskou et al. for unknown pieces of source code. They first select the L most frequent n-grams to generate a profile set and then define a similarity measure between two different profile sets by the size of the intersection of these two sets. Their method has been tested against 267 programs in C++ by 6 different programmers. One half of the data are used as training data and the other half as testing data. The results show that the accuracy can reach 100% with individual n-grams. For programs written in Java, its accuracy is up to 97% (Frantzeskou et al., 2006).

5.2.2. User profiles in plagiarism detection

Authorship determination does not only enable associating a name with a tract of otherwise-anonymous text, but can also be used to determine when a tract of text was written by someone other than the putative author. The widespread use of computers and the advent of the Internet have made it easier to plagiarise the work of others. Plagiarism is not only found in academia, but can also can be found in scientific papers, art designs, and source code. Many detection methods exist for plagiarism, which can be either manual or automatic. As manual detection requires substantial effort and memory, it is not always practical. However, automatic detection allows vast collections of documents to be compared to each other, so it becomes more likely to make the detection successful (Plagiarism detection).

Stamatatos (2009) presents this technique to check intrinsic plagiarism. Specifically, he uses a sliding window moving along the whole text. Each time he compares the text in the window with the entire text. He also applies heuristic rules for this plagiarism detection. In his experiment, 3-grams are used, the window width was set to 1000 characters, and the step for a sliding window was set to 200 characters. He selected more than 3,000 texts whose lengths varied from 3,000 to 2.5 million characters. The result yielded a 78% accuracy at confirming no plagiarism on plagiarism-free documents. The method, however, can be affected by the amount of plagiarism, as it detects fragments of plagiarism against a background of non-plagiarised material, so if there is too much plagiarism, it will change the writing style of the text.

In order to achieve a better performance, Barron-Cedeno et al. (2010) conduct an analysis experiment and automatically detect shared contents in written documents for text reuse and plagiarism. In their pre-processing procedure, they first replace all non-alphabetic terms with words of the same length (the maximum of length is 9). They make a similarity estimation by these simple n-grams. The results show that their approach does not much affect the detection accuracy of the retrieval process. Their methods claim to significantly reduce the computational time and storage. Obviously, useful information is lost in this approach, which is not suitable for use in the analysis of small datasets.

Abou-Assaleh et al. (2004) conducted an n-gram approach for detecting new malicious code. They generated n-gram signatures from collections of malicious code and benign code and then classified an unseen code against these signatures. They adapted the Common n-gram method and selected the L most frequent n-grams. Their classification criteria were based on the method of k-nearest neighbours. The data included 25 worms and 40 healthy program files. By carefully configuring the relevant parameters during training, the training accuracy can be 95% and its 5-fold cross-validation average accuracy can reach 94%.

5.2.3. User profiles in Astroturfing detection

Authorship determination also can be used to match groups of texts as well as introducing the use of metadata for this purpose. The use of so-called "astroturfing" is widespread, from business to politics, from army to civil, and from book reviews to online surveys. It is defined as the practice of masking the sponsors of a message or organisation to make it appear as though it originates from and is supported by grassroots participant(s) (Astroturfing). It is a practice intended to give the statements or organisations enhanced credibility by withholding information about the source's financial connections, as it is known that people are less inclined to trust messages which they feel are financially- or politically-motivated. Astroturfing is also an attempt to create an impression of widespread grassroots support for a policy, individual, or product, where in fact little such support exists. As the software improves, they will become even more difficult to detect (Bienkov, 2012). These astroturfers are well-known as 'internet water army' in China ('water' here means 'hidden' in Chinese) and they generally work for profits (Chen et al., 2013).

Clearly there is scope for authorship attribution methods to work here. However instead of aiming to attribute a text to one of a set of known authors or determine multiple authorship (as in plagiarism), in astroturfing detection it is a case of determining when a single author appears to lie behind numerous handles (online names). The question to ask here is whether a collection of

putatively different people are in fact the same agent. Another question to ask is whether any of the supposed human agents fit the pattern of non-human agents (bots). This latter case is similar in principle to the malware detection use described above, as it analyses text to see if the writer falls into a certain class of user, rather than matching any specific user.

Ratikiewicz et al. (2011a) explore the interaction behaviours between individuals in online social media by analysing the 'meme' diffusion from the twitter 'Gardenhose', which includes analysing 'meme' network by Klatsch framework and sentiment by a modified version of the Google-based Profile Mood States. They use AdaBoost and support vector machines as the classifiers for their experiment. The accuracy is around 90% with false negative rate of at least 5%. Later, they employ the similar methods to analyse the behaviours of astroturfers for political purposes on social media. The extracted features from Twitter are topological, content-based and crowd-sourced information. Finally they identify early astroturfers by machine learning techniques. The result of 96% detection accuracy in 2010 U.S. midterm elections show their methods are promising (Ratkiewicz et al., 2011b).

Chen et al. (2013a, 2013b) present an effective detection approach for classifying hidden paid posters on social networks. By analysing the profiles of paid posters, they describe their behaviours by follows: percentage of replies, average interval time of posts, active period, and others. Their analysis methods are both statistical and semantic. They gather data from 745 users on 2 websites for about 3 months. The results show the accuracy can be 63% with only the 4 types of statistical features involved, but the accuracy can increase up to 88% if the semantic features added together.

In summary, it can be seen that psychometric profiles, in particular the analysis of written text, can be an effective tool for identifying users, and has the potential to contribute significantly to user validation and intrusion detection.

5.3. Combining profiles in IDS

Anomaly-based detection based on a single profile is adopted to profile a user's behaviour in some research (Yeung and Ding, 2002; Forrest et al., 1996; Cabrera et al., 2001; McKinney and Reeves, 2009; Lane and Brodley, 1997), but multiple, combined profiles are more often used so as to enable better profiling and more accurate diagnosis (Blasing et al., 2010; Marceau, 2000; Anderson, 1980; Li et al., 2012; Tabia and Benferhat, 2008; Stein et al., 2005, 2010; Mazzariello and Oliviero, 2006; Abou-Assaleh et al., 2004; Pannell and Ashman, 2010). Sometimes, it can also avoid wrongly excluding a user if they have a genuine alteration in a specific profile, for example, a user who has injured an arm may type into a keyboard very differently. Many IDSs integrate several performance profiles, such as CPU usage and memory usage.

There are different combinations of user and system profiles in the related literatures. Some put several profiles of system behaviours together for analysis as the combination of profiles can gain overall better performance than individual profiles. Due to this merit, there have been more attempts recently on this area such as analysis of the difference, in terms of speed or accuracy, between using a singular profile and combining multiple profiles. For example, Anderson (1980) combines session logs, device usage, file usage durations and program usage to indirectly profile user behaviours; Denning (1987) detects anomalies by extracting user logins, CPU usage and I/O usage. Others combine different user behaviours directly for anomaly detections. Bergadano et al. (2003) consider the latency, duration and order of keystrokes for profiling users' behaviours, but Qiu and Cho (2006) extract user profiles by users' higher properties together by combining their interests and preferences. Moreover, Vizer et al. (2009) combine the low-level user profiles of keystrokes with high-level ones of linguistic features (lexical and syntactic). Further, Pannell and Ashman (2010) model user behaviours by combining both system profiles (memory usage, CPU usage and window usage) and user data (keystrokes).

Jagadeesan and Hsiao (2009) use an interaction ratio for the mouse and keyboard and show that the re-authentication system obtained 96.4% accuracy. Kandias et al. (2010) proposed an attack and predication model based on psychological data. The proposed model contained two types of user information: one was the user's psychological description; the other was the usage information of information systems for a specific user. The psychological description included three steps: first, a user's sophistication level was determined; then malicious behaviours and finally stress levels of users were evaluated. For preference, the following questionnaires were used: past misdeeds; ability to imitate other people's ideas; influence of family and friends; difference association; penalties awareness and reward-and-penalty balance; moral failure; public service awareness; blaming victims or demeaning. The user's stress levels were first obtained through psychometric tests, which included both a personal stress and an occupational one.

A summary of profiles adopted in the literatures can be found in Table 1.

6. Conclusion and future opportunities

We reviewed the related literature on intrusion detection and prevention systems from the viewpoint of exploiting the behaviour of the user in the context of their user profile to confirm or deny the legitimacy of their presence on the system. We briefly introduced IDS, prior to categorising behaviours into system behaviours and user behaviours as well as classifying profiles into different classes and subclasses such as behavioural biometrics and psychometrics. Finally, we explored applications of combining user profiles on IDS.(Table 2).

From the literature reviewed in this paper, it is evident that there is substantial research on anomaly-based intrusion detection. The reviewed methods are used for user authentication, authorship attribution, plagiarism detection, deception detection, astroturfing, among others. Some researchers use system profiles to profile user behaviours by observing system behaviours such as system calls, usage of programs, devices and files, shell commands, network status, and so on. Others take advantage of user behaviours directly; for example, keystrokes, n-grams, lexical and syntactic features, among others. A wide range of analysis techniques have beenapplied to these profiles. The advantages and disadvantages of each are summarised in Table 3. The analysis methods in the summary are not exclusive, as some works only mention the general techniques adopted, such as data mining and machine learning, but others name the specific methods such as C4.5 and the finite state machine. We summarised the relevant literature based on the analysis method(s) adopted, the IDS type it belongs to, the types of behaviours handled, the types of profiles it analyses, and the advantages and disadvantages (Peng et al., 2016).

Many detection analysis methods are used for detecting anomaly behaviours, some use statistical approaches (Denning, 1987; Anderson, 1980; Pannell and Ashman, 2010; Chebrolua et al., 2005; Maxion, 2003; Pannell and Ashman, 2010; McKinney and Reeves, 2009; Martin et al., 2005; Jagadeesan and Hsiao, 2009; Fox et al., 2012; Cavnar and Trenkle, 1994; Wressnegger et al., 2013; Li

Table 1
Summary of profiles used in the literature review.

Profiles	References
System usages	Blasing et al. (2010), Marceau (2000), Tabia and Benferhat (2008), Denning (1987), Li et al. (2012), Anderson (1980), Forrest et al. (1996), Li et al. (2006), Chen et al. (2013), Pannell and Ashman (2010)
Network usages	Stein et al. (2005), Tabia and Benferhat (2008), Mazzariello and Oliviero (2006), Li et al. (2012), Ratkiewicz et al. (2011), Chen et al. (2013)
System calls	Blasing et al. (2010), Forrest et al. (1996), Cabrera et al. (2001), Li et al. (2012), Li et al. (2006)
User phone calls	Cortes and Pregibon (2001)
User logins	Denning (1987), Chen et al. (2013)
Virus signatures	Venugopala and Hu (2008)
Biometric	Bhaskaran et al. (2011)
GUI related	Li et al. (2006), Pannell and Ashman (2010),
Commands	Blasing et al. (2010), Yeung and Ding (2002), Li et al. (2012), Lane and Brodley (1997)
Keystrokes	Alexandre (1997), Vizer et al. (2009), Umphress and Williams (1985), Bergadano et al. (2003), Revett (2009), Pannell and Ashman (2010)
Lexical	Afroz et al. (2012), Barron-Cedeno et al. (2010), Shrestha and Solorio (2013), Houvardas and Stamatatos (2006), Vizer et al. (2009), Frantzeskou et al. (2006), Iqbal et al. (2008), Stein et al. (2010), Hirst and Feiguina, (2007), Abou-Assaleh et al. (2004), Ratkiewicz et al. (2011), Fox et al. (2012), Masud et al. (2007), Cavnar and Trenkle (1994)
Syntactic	Afroz et al. (2012), Barron-Cedeno et al. (2010), Shrestha and Solorio (2013), Houvardas and Stamatatos (2006), Vizer et al. (2009), Frantzeskou et al. (2006), Iqbal et al. (2008), Stein et al. (2010),Hirst and Feiguina (2007), Abou-Assaleh et al. (2004), Ratkiewicz et al. (2011), Fox et al. (2012), Masud et al. (2007), Cavnar and Trenkle, (1994)
Content-specific	Afroz et al. (2012), Barron-Cedeno et al. (2010), Shrestha and Solorio (2013), Houvardas and Stamatatos, (2006), Vizer et al. (2009), Frantzeskou et al. (2006), Iqbal et al. (2008), Stein et al. (2010), Hirst and Feiguina (2007), Abou-Assaleh et al. (2004), Ratkiewicz et al. (2011), Fox et al. (2012), Masud et al. (2007), Cavnar and Trenkle (1994)
User interests/preferences	Chen et al. (2013), Qiu and Cho (2006)

et al., 2005; Pannell and Ashman, 2010; Bailey et al., 2014; Hovold, 2005), artificial neural networks (Li et al., 2006; Vizer et al., 2009; Jagadeesan and Hsiao, 2009), hidden Markov models (Yeung and Ding, 2002; Li et al., 2012; Huang and Stamp, 2011), rule learning (Revett, 2009; N., B.F. M., 2011; Lane and Brodley, 1997), decision trees (Afroz et al., 2012; Li et al., 2006; Tabia and Benferhat, 2008; Stein et al., 2005; Chebrolua et al., 2005; Stein et al., 2010), support vector machines (Afroz et al., 2012; Li et al., 2006; Ratkiewicz et al., 2011; Chen et al., 2013; Hirst and Feiguina, 2007; Yang, 2010; Abbasi and Chen, 2008) and fuzzy computing (Chebrolua et al., 2005; Biermann, 2001). In Table 3, we summarised the analysis techniques and their advantages and disadvantages.

Although behaviour-based approaches can detect new intrusions with less dependence on operating system-specific mechanisms, their high false positiverate is generally cited as the main drawback of behavior-based techniques. This is mainly either because the entire scope of the behavior of an information system may not be covered during the learning phase, or because the behaviors can legitimately change over time. Therefore the IDS needs periodic retraining of the behavior profile, or perhaps even ongoing training, resulting in updating behavior models for detection. There is, however, a risk attached with ongoing training that an attacker could over time manipulate the profile away from the genuine user into that of the attacker.

The profiles extracted from the data are diverse: from basic ones such as numbers of clicks and characters to advanced features such as semantic dependencies and language-specific features (Yeung and Ding, 2002; Wressnegger et al., 2013; Pannell and Ashman, 2010; Yang, 2010; Koong et al., 2014; Yang and Fang, 2013; Yang and Padmanabhan, 2010; M., C. and M. G., 2011; Masri et al., 2014). In Table 4, we categorised the literature based on IDS types, behaviours and profiles.

Although their detection analysis methods differ widely, the procedures are much the same. There are two phases: one is the learning phase for building legitimate detection model with permitted users' data, and the other is the operational phase for detecting anomalies against the model in real-time.

Generally, we can see the simple analysis methods can achieve better time performance at the cost of relatively lower detection rates (Umphress and Williams, 1985; Forrest et al., 1996; Bergadano et al., 2003; Maxion, 2003; Cavnar and Trenkle, 1994; Hovold, 2005), while more sophisticated methods can have high accuracy of detection but can be time-consuming (Li et al., 2012; Chebrolua et al., 2005; Martin et al., 2005). Also, it is noticed that psychometric analysis approaches often appear to produce a higher detection rate and lower error rate than behavioural biometric ones, but may be harder to process and have poorer performance in time and space (Afroz et al., 2012; Pannell and Ashman, 2010; Ratkiewicz et al., 2011; Ratkiewicz et al., 2011; Beghdad, 2004).

There still remain some unsolved questions in the use of behavioural profiles in intrusion detection. While individual profiles may have demonstrated efficacy, they are individually susceptible to false positives when the genuine user for some reason is not behaving in their normal fashion (e.g. keystroke timings with an injured arm).Also, some may be susceptible to exploitation by attackers who input false values, if there is no reliable validation of the input data. Some profiles may work only over certain types of data and not others (e.g. mouse use). So clearly a combination of profiles would be desirable, reducing false positives and ensuring all forms of complementary user behaviour can be integrated into the intrusion detection process. This then leads to the need to not just understand which behavioural profiles should form part of the user profile, but what priority should apply, and what combinations are sufficient to raise an alarm.

The robustness of behavioural profiles as intrusion detection features also needs further investigation. It is helpful to measure accuracy, including false positive rates, but false negative rates need more focus as well. In particular, each form of behavioural profile may be compromised by attackers if they should acquire a copy of the user profile information. Of course, the user profile should, where possible, be kept private, but some aspects of user behaviour are impossible to hide. For example, a user who frequently posts on social media will have a profile writing style that any attacker reading those posts could use to form their own profile of that user. Other data such as keystrokes could also be captured as users enter text into public sites (allowing for lag time).

Hence, the question then becomes not one of how to keep the user profile secret, which is all but impossible, but how to ensure that profiles are chosen to be hard for an attacker to falsify to match that of the genuine user. This depends partly on the trustworthiness of the source of that information. As noted above, if keystroke data is

Table 2
Summary of literature review.

Author	Methods	Types	Behaviours	Profiles	Advantages	Disadvantages
Anderson (1980)	S	A	S	H	Simple algorithm; Fast	Inflexible; Low performance (High FP and FN)
Alexandre (1997)	ANN	S	U	B	High accuracy; Fast; Easy implementation	Cannot detect new types of user behaviours
Umphress and Williams (1985)	S	A	U	B	Simple algorithm; Fast, High performance (FP:6%)	Single profile; High false positive; Inflexible
Denning (1987)	AR	A	S	H	Independent model; Wide range of intrusions	No implementation; High false positive; Inflexible
Cortes and Pregibon (2001)	S	S	U	P	High accuracy; Fast; Easy implement	Cannot detect new type of user behaviours
Cavnar and Trenkle (1994)	N-gram	A	U	P	Efficiency; Satisfactory performance	Formal language; Only select most n-grams with higher frequency; Profiles are not normalized
Forrest et al. (1996)	PR	A	S	H	Different method from previous ones; Simple algorithm; Efficiency	Low performance; Hard to set a proper short-range Single profile; High false positive rate
Lane and Brodley (1997)	ML	A	S	H	High performance (Detection rate:97%)	Complicated algorithm; Dataset is specialized; High false positive rate; Single profile; Inflexibility
Marceau (2000)	FSM	A	S	H	Pragmatic	Low performance rate; High false positive rate; Single profile
Cabrera et al. (2001)	PR	A	S	H	Hybrid method	Low performance (Detection rate:75%); High false positive rate; Single profile
Yeung and Ding (2002)	SDM	A	S	H	Simple; Efficient	Low performance; High false positive rate
Bergadano et al. (2003)	SDM	A	U	B	High performance (Accuracy:90%)	Simple algorithm; High false positive rate; Inflexible
Abou-Assaleh et al. (2004)	KNN, N-gram	A	U	P	High performance (Accuracy: 90%)	Only select L most freq n-grams; Formal language; Similar topics
Stein et al. (2005)	DT, GA	A	S	N	Hybrid; High classifying accuracy	Complicated algorithm; High false positive rate
Venugopala and Hu (2008)	PR	S	S	H	High time performance (50% less than Clam-AV)	Relatively low detection rate
Li et al. (2006)	ANN, SVM	A	S	H	Satisfactory performance; Less training data	Complicated algorithm; High false positive rate; Inflexible
Mazzariello and Oliviero (2006)	ANN	A	S	N	Self-aware and self-organizing; combining several analysis methods	No implementation; High false positive rate; Inflexible
Qiu and Cho (2006)	SDM, S	A	U	P	More reliable than PageRank; Efficient	Simple algorithm; High false positive rate
Frantzeskou et al. (2006)	N-gram	A	U	P	High performance (Accuracy: 95%)	Formal language; Similar topics
Houvardas and Stamatatos (2006)	SVM, N-gram	A	U	P	Large scale data and authors	Similar topics; Formal language; Lower accuracy: 72%
Masud et al. (2007)	SVM, N-gram	A	U	P	High performance; Accuracy: 96% with FP/FN: 5.4/2.6	Formal language; Similar topics
Hirst and Feiguina (2007)	SVM, N-gram	A	U	P	For short text; Accuracy: from 95%～99%; dependent on # of block size	Formal language; Similar topics
Tabia and Benferhat (2008)	RML, C4.5	A	S	N	Less memory; Higher detection rate	Complicated algorithm; High false positive rate; Inflexible
Iqbal et al. (2008)	DM, N-gram		U	P	Accuracy: 70–90%	Formal language; Similar topics
Revett (2009)	PR	A	U	B	High performance (FN:0.0%,FP:0.8%); Cheaper (software) implementation	Fewer attributes; Simple scoring scheme; High false positive rate; Inflexible
Vizer et al. (2009)	SVM, ANN	A	U	B, P	Low cost; Satisfactory performance	Complicated algorithms; High false positive rate; Inflexible
McKinney and Reeves (2009)	BA	A	S	H	Satisfactory performance	Complicated algorithms; High false positive rate; Inflexible
Pannell and Ashman (2010)	S	A	S, U	H, B	Easy to implement with both biometric and psychometric; Satisfactory performance (90 s)	Additional reliable features; More sophisticated algorithms; High false positive rate
Blasing et al. (2010)	SDM	S	S	H	High accuracy; Fast; Easy implement	Cannot detect new type of user behaviours
Stein et al. (2010)	SVM, N-gram	A	U	P	Precision: 72%～98%	Formal language(artificially plagiarized; documents); Similar topics
Barron-Cedeno et al. (2010)	S, N-gram	A	U	P	efficient	Gain speed at cost of accuracy; Formal language(Wikipedia);Similar topics
Ratkiewicz et al. (2011)	DM	A	S	N	Distributed, large scale	Resource-consuming
Bhaskaran et al. (2011)	ML, BA	A	U	B	Satisfactory accuracy: 82.5%	Complicated algorithm; High false positiverate; Inflexibility; Extra device
Li et al. (2012)	HMM	A	S	H	Efficiency; High performance (detection rate: 96.93%)	Complicated algorithm; High false positive rate; Inflexibility
Fox et al. (2012)	N-gram	A	U	P	Efficiency; Satisfactory performance	Complicated algorithm; High false positive rate; Inflexibility
Afroz et al. (2012)	SVM	A	U	P	Accuracy: 96.6%	Most are formal language and only one of their 3 data sets is from blogs; Similar topics
Shrestha and Solorio (2013)	N-gram	A	U	P	Precision: 88–99%	Formal language; Similar topics
Chen et al. (2013)	S, SVM	A	U	P	Precision: 95.24%	Similar topics

[**Methods**: **S**-Statistics; **ANN**-Artificial Neural Network; **PR**-Pattern Recognition; **ML**-Machine Learning; **DM**-Data Mining; **FSM**-Finite State Machine; **SDM**-Static/Dynamic Model; **DT**-Decision Tree; **GA**-Genetic Algorithm; **SVM**-Support Vector Machine; **RML**-Relaxing minimum Description Length principle; **BA**-Bayes Algorithm; **AR**-Associate Rule; **HMM**-Hidden Markov Model.
IDS Types: **S**-Signature based IDS; **A**-Anomaly based IDS.
Behaviours: **S**-System behaviour; **U**-User behaviour.
Profiles: **H**-Host based system profile; **N**-Network based system profile; **B**-Biometric user profile; **P**-Psychometric user profile.]

Table 3
Analysis algorithms and their advantages and limitations.

Techniques	Advantages	Disadvantages
Artificial Neural Network	Non-parametric and nonlinear model; Easily implemented in parallel architectures; Data-driven, self-adaptive, flexible; High degree of accuracy; Suitable for large amount of data sets;	Difficult to pick the correct topology; Long time training with large data; Output/issues are incomprehensible; Slow to converge; Hard to set parameters and interpret; Cannot deal with uncertainties;
Hidden Markov Model	Effective; Can handle variations in record structure.	Training using annotated data Not completely automatic May require manual markup Size of training data may be an issue.
Association Rule	Easy to interpret and explain(understand); High performance; Suitable for large item set property; Easily parallelized and implemented; Best suitable for categorical data analysis.	Requires many database scans; Trivial rules produced and poorly understandable; No expressions of cause and effect.
N-Gram	Encode both keywords and word ordering; Whether Models are biased or not is dependent on real data; Fast and easy to learn features of each affect type.	Unable to capture long range dependencies; Dependent on having a corpus of data to train from.
K-Nearest Neighbour	Work well for lower dimensions; Robust to noisy training data; Robust with regard to the search space; Few parameters to tune; Simple implementation.	Unsuitable for very higher dimensions; Need to determine value of parameter k; High computational cost; Large storage requirements; Expensive testing of each instance; Sensitive to noisy, irrelevant attributes and very unbalanced datasets.
Decision Tree	Easy to interpret and explain; Non-parametric(outlier); Fast to train and easy to evaluate and interrupt.	Easily data-overfitting; May need ensembles to help reduce variance; Hard to update model; Unsuitable for numerical values and need to converse to nominal values;
Naive Bayes	Very simple representation; Less training data; Converge quickly; Easy to understand and implement; High performance.	Doesn't allow for rich hypotheses; Too constrictive assumption of attribute independence; Unsuitable for unbalanced classes; Only for nominal attributes;
Support Vector Machine	High accuracy; No overfitting; Suitable for large amounts of data; Regularized parameter for avoiding over-fitting; Build in expert knowledge of problem; Being an approximation to a bound on the test error rate.	Many parameter tuning; Hard to pick/find the right kernel; Incomprehensible results/output; No standardized way for dealing with multi-class problems; Hard to update model; Computationally expensive(slow performance);

being sent from a user's device, it cannot be used to validate that user as the user may have compromised the keystroke data feed. If the user was operating an app that altered the keystroke data in real time, all that an attacker would require would be access to the keystroke data from a legitimate user's profile to then adjust their own keystroke data to match. Therefore, for this reason alone, keystroke data, however rapidly it identifies intruders, needs to be supplemented with complementary profiles because it is too simple to falsify the input data.

So how easy would it be to falsify input data to fool the intrusion detection system for a given behavioural profile? This is the next step for research into behavioural profiling, to establish how susceptible each of the selected profile characteristics is to falsification. We posit here that psychometric profiles are more difficult to mimic than biometric profiles because they are more challenging to accurately model, and are not as easily reducible to numeric values. Certainly, in, say, text analysis, it is possible to count word frequencies, and tally misspellings and grammatical errors, and even look at structure of text, but more profile text comprehension is computationally demanding. Even knowing such information does not help automatically change an attacker's writing style, especially in real time. In fact it is hard to mimic a writing style even without time constraints, and efforts to disguise writing have not always been successful, as investigation into authorship of Shakespeare's plays attest. However the conjecture that psychometric data is harder to falsify is not yet supported with evidence, so trials need to be conducted to determine how easily and how accurately people can write in a style that mimics another person accurately enough to avoid detection.

In closing, we note that behavioural profiling shows promise for use in intrusion detection. Additional research will establish the parameters of its use and confirm its benefits.

Table 4
Summary of methods, behaviour and profiles.

	Types		Behaviours		Profiles	
	Signature-based	**Anomaly-based**	**System**	**User**	**Biometric**	**Psychometric**
References	Alexandre (1997), Cortes and Pregibon (2001), Venugopala and Hu (2008), Blasing et al. (2010)	Afroz et al. (2012), Barron-Cedeno et al. (2010), Marceau (2000), Shrestha and Solorio (2013), Houvardas and Stamatatos (2006), Yeung and Ding (2002), Stein et al. (2005), Vizer et al. (2009), Tabia and Benferhat (2008), Umphress and Williams (1985), Forrest et al. (1996), Bergadano et al. (2003), Denning, (1987), Cabrera et al. (2001), Mazzariello and Oliviero (2006), Li et al. (2012), Anderson (1980), McKinney and Reeves (2009), Revett (2009), Li et al. (2006), Bhaskaran et al. (2011), Lane and Brodley (1997), Frantzeskou et al. (2006), Iqbal et al. (2008), Stein et al. (2010), Hirst and Feiguina (2007), Abou-Assaleh et al. (2004), Ratkiewicz et al. (2011), Chen et al. (2013), Pannell and Ashman (2010), Cavnar and Trenkle (1994), Fox et al. (2012), Qiu and Cho (2006), Masud et al. (2007)	Venugopala and Hu (2008), Blasing et al. (2010), Marceau (2000), Yeung and Ding (2002), Stein et al. (2005), Tabia and Benferhat (2008), Forrest et al. (1996), Bergadano et al. (2003), Cabrera et al. (2001), Mazzariello and Oliviero (2006), Li et al. (2012), Anderson (1980), McKinney and Reeves (2009), Li et al. (2006), Lane and Brodley (1997), Stein et al. (2010), Abou-Assaleh et al. (2004), Pannell and Ashman (2010)	Alexandre (1997), Cortes and Pregibon (2001), Afroz et al. (2012), Barron-Cedeno et al. (2010), Shrestha and Solorio (2013), Houvardas and Stamatatos (2006), Vizer et al. (2009), Umphress and Williams (1985), Denning (1987), Mazzariello and Oliviero (2006), Anderson (1980), Revett, (2009), Bhaskaran et al. (2011), Lane and Brodley, (1997), Frantzeskou et al. (2006), Iqbal et al. (2008), Hirst and Feiguina, (2007), Ratkiewicz et al. (2011), Chen et al. (2013), Cavnar and Trenkle (1994), Fox et al. (2012), Qiu and Cho (2006), Masud et al. (2007)	Alexandre (1997), Cortes and Pregibon (2001), Venugopala and Hu (2008), Blasing et al. (2010), Marceau, (2000), Yeung and Ding (2002), Stein et al. (2005), Vizer et al. (2009), Tabia and Benferhat (2008), Umphress and Williams (1985), Forrest et al. (1996), Bergadano et al. (2003), Denning (1987), Cabrera et al. (2001), Li et al. (2012), Anderson (1980), McKinney and Reeves, (2009), Revett (2009), Li et al. (2006), Bhaskaran et al. (2011), Pannell and Ashman (2010)	Afroz et al. (2012), Barron-Cedeno et al. (2010), Shrestha and Solorio (2013), Houvardas and Stamatatos (2006), Frantzeskou et al. (2006), Iqbal et al. (2008), Stein et al. (2010), Hirst and Feiguina (2007), Abou-Assaleh et al. (2004), Ratkiewicz et al. (2011), Chen et al. (2013), Cavnar and Trenkle (1994), Qiu and Cho (2006), Masud et al. (2007)

Acknowledgements

The authors thank the editor and the three anonymous reviewers for their constructive and generous feedback.

References

Abbasi, A., Chen, H., 2008. Writeprints: a stylometric approach to identity-level identification and similarity detection in cyberspace. ACM Trans. Inf. Syst. 26 (2), 1–29.
Abdel-Hafez, A., Xu, Y., 2013. A survey of user modelling in social media websites. Comput. Inf. Sci. 6 (4).
Abou-Assaleh, T., et al., 2004. Detection of new malicious code using N-grams signatures. In: Proceeding of Second Annual Conference on Privacy, Security and Trust, October: pp. 13–15.
Afroz, S., Brennan, M., Greenstadt, R., 2012. Detecting hoaxes, frauds, and deception in writing style online. In: Proceedings of the IEEE Symposium on Security and Privicy (Sp) pp. 461–475.
Alexandre, T.J., 1997. Biometrics on smart cards: an approach to keyboard behavioral signature. Future Gener. Comput. Syst. 13, 19–26.
Anderson, J.P., 1980. Computer Security Threat Monitoring and Surveillance. James P. Anderson Company, Fort Washington, PA.
Astroturfing, ⟨https://en.wikipedia.org/wiki/Astroturfing⟩.
Bailey, K.O., Okolica, J.S., Peterson, G.L., 2014. User identification and authentication using multi-modal behavioral biometrics. Comput. Secur. 43, 77–89.
Barron-Cedeno, A., et al., 2010. Word length n-grams for text re-use detection. Comput. Linguist. Intell. Text Process., 687–699 2010: p. 687-699.
Beghdad, R., 2004. Modelling and solving the intrusion detection problem in computer networks. Comput. Secur. 23 (8), 687–696.
Bergadano, F., Gunetti, D., Picardi, C., 2003. Identity verification through dynamic keystroke analysis. Intell. Data Anal 7 (5), 469–496 7(5): p. 469-496.
Bienkov, A., Astroturfing: what is it and why does it matter? ⟨http://www.theguardian.com/commentisfree/2012/feb/08/what-is-astroturfing⟩, 2012.
Biermann, E., 2001. A comparison of Intrusion Detection systems. Comput. Secur. 20, 676–683.
Blasing, T., et al., 2010. An android application sandbox system for suspicious software detection. In: Proceedings of the 5th International Conference on Malicious and Unwanted Software (Malware 2010) (MALWARE'2010), Nancy, France, France.
Brusilovsky, P., 2001. Adaptive Hypermedia. User Modeling and User-Adapted Interaction, Vol. 11, no. 1, pp. 87–110.
Cabrera, J.B.D., Lewis, L., Mehra, R.K., 2001. Detection and classification of intrusions and faults using sequences of system calls. SIGMOD Rec. 30 (4), 25–34.
Casas, P., Mazel, J., Owezarski, P., 2012. Unsupervised network intrusion detection systems: detecting the unknown without knowledge. Comput. Commun. 35 (7), 772–783.
Cavnar, W.B., J.M. Trenkle, 1994. N-gram-based text categorization. In Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval, SDAIR-94, pp. 161–175.
Chandola, V.A., Banerjee, Kumar, V., 2009. Anomaly detection. ACM Comput. Surv. 41 (3), 1–58.
Chandola, V.,B.A., Kumar, V.,, 2012. Anomaly detection for discrete sequences: a survey. IEEE Trans. Knowl. Data Eng. 24 (5), 823–839.
Chebrolua, S., Abrahama, A., Thomas, J.P., 2005. Feature deduction and ensemble design of intrusion detection systems. Comput Secur 24, 295–307.
Chen, C.M., et al., 2013. Battling the internet water army: detection of hidden paid posters. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM'13, pp. 116–120.
Chen, C.M., et al., 2013. Battling the internet water army: detection of hidden paid posters. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM'13, pp. 116–120.
Cortes, C., Pregibon, D., 2001. Signature-based methods for data streams. Data Min. Knowl. Discov. 5, 167–182.

Davis, J., Clark, A., 2011. Data preprocessing for anomaly based network intrusion detection. A Rev. Comput. Secur. 30, 353–375.
Denning, D.E., 1987. An intrusion detection model. IEEE Trans. Softw. Eng. SE-13 (2), 222–233.
Eugene, S., 2008. An information security pioneer. IEEE Secur. Priv. 6 (1), 9.
F., Q. and C. J., 2006. Automatic identification of user interest for personalized search. In: Proceedings of the 15th Intnational World Wide Web Conference, 2006: pp. 727–736.
Forrest, S., et al., 1996. A sense of self for Unix processes. In: Proceedings of the IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, 1996: pp. 120–128.
Fox, N., Ehmoda, O., Charniak, E., 2012. Statistical Stylometrics and the Marlowe-Shakespeare Authorship Debate.
Frantzeskou, G., et al., 2006. Effective identification of source code authors using byte-level information. In: Proceedings of the 28th International Conference on Software engineering, ICSE'06, pp. 893–896.
Gates, C. and C. Taylor, 2006. Challenging the anomaly detection paradigm: a provocative discussion. In: Proceedings of the 2006 Workshop on New Security Paradigms, ACM, Germany, pp. 21–29.
Han, H., X.-l. Lu, and L.-y. Ren, 2002. Using data mining to discover singatures in network-based intrusiton detection. Proceedmgs of the First International Conference on Machme Learmng and Cybemeucs, Beqmg, 4–5 November 2002.
Hirst, G., Feiguina, O., 2007. Bigrams of syntactic labels for authorship discrimination of short texts. Lit. Linguist. Comput. 22 (4), 405–417.
Houvardas, J., Stamatatos, E., 2006. N-gram feature selection for authorship identification AIMSA 2006, LNCS(LNAI). 4183, 77–86.
Hovold, J., 2005. Naive Bayes spam filtering using word-position-based attributes. In: Proceedings of the Second Conference on Email and Anti-spam, CEAS, Stanford University.
⟨http://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/⟩.
Huang, L., Stamp, M., 2011. Masquerade detection using profile hidden Markov models. Comput. Secur. 30 (8), 732–747.
Iqbal, F., et al., 2010. Mining writeprints from anonymous e-mails for forensic investigation. Digit Investig. 7 (1–2), 56–64.
Iqbal, F., et al., 2008. A novel approach of mining write-prints for authorship attribution in e-mail forensics. Digit Investig. 5, S42–S51.
Jacob, G., Debar, H., Filiol, E., 2008. Behavioral detection of malware: from a survey towards an established taxonomy. J. Comput. Virol. 4 (3), 251–266.
Jagadeesan, H. and M.S. Hsiao, 2009. A Novel Approach to Design of User Re-Authentication Systems.
Jin, L., et al., 2013. Understanding user behavior in online social networks: a survey. IEEE Commun. Mag., 143–150.
Jøsang, A., Ismail, R., Boyd, C., 2007. A survey of trust and reputation systems for online service provision. Decis. Support Syst. 43 (2), 618–644.
Kandias, M., et al., 2010. An insider threat prediction model. Trust, Priv. Secur. Digit Bus. 6264, 26–37.
Keselj, V., et al., 2003. N-gram-based author profiles for authorship attribution. In: Proceedings of the Pacific Association for Computational Linguistics, pp. 255–264.
Khan, S., Gani, A., Abdul Wahab, A., Bagiwa, M., Shiraz, M., Khan, S., Buyya, R., Zomaya, A., 2016. Cloud log forensics: foundations, state of the art, and future directions. ACM Comput. Surv. 49 (1), Article 7.
Kheyri, D., Karami, M., 2012. A comprehensive survey on anomaly-based intrusion detection in MANET. Comput. Inf. Sci. 5, 4.
Koong, C.S., Yang, T.I., Tseng, C.C., 2014. A user authentication scheme using physiological and behavioral biometrics for multitouch devices. Sci. World J. 2014, 781234.
Koppel, M., Schler, J., Argamon, S., 2009. Computational methods in authorship attribution. J. Am. Soc. Inf. Sci. Technol. 60 (1), 9–26.
Lane, T. and C.E. Brodley, 1997. An application of machine learning to anomaly detection. In: Proceedings of the 20th National Information Systems Security Conference, pp. 366–380.
Li, L., S. Sui, and C.N. Manikopoulos, 2006. Windows NT User profiling for masquerader detection. In: Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control, ICNSC'06. pp. 386–391.
Li, X., Y. Xue, and B. Malin, 2012. Detecting anomalous user behaviors in workflow-driven web applications. In: Proceedings of the 31st International Symposium on Reliable Distributed Systems (SRDS 2012), pp. 1–10.
Li, W.-J., et al., 2005. Fileprints: identifying file types by N-gram analysis. In: Proceedings of the 2005 IEEE Workshop on Information Assurance and Security, pp. 64–71.
Corney, M., Mohay, G., 2011. Detection of anomalies from user profiles generated from system. In: Proceeding AISC'11 Proceedings of the Ninth Australasian Information Security Conference, 116: pp. 23–32.
Maor, E., ⟨https://securityintelligence.com/behavioral-profiling-finding-man-wasnt/⟩, April 17, 2013.
Marceau, C., 2000. Characterizing the behaviour of a program using multiple-length N-grams. In: Proceedings of the 2000 Workshop on New Security Paradigms, ACM, Ballycotton, County Cork, Ireland, pp. 101–110.
Martini, B., Choo, K.-K.R., 2014. Distributed filesystem forensics: XtreemFS as a case study. Digit Investig. 11 (4), 295–313.
Martin, S., et al., 2005. Analyzing behaviorial features for email classification. In: Prodeedings of the IEEE Second Conference on Email and Anti-Spam (CEAS 2005).
Masri, W., Assi, R.A., El-Ghali, M., 2014. Generating profile-based signatures for online intrusion and failure detection. Inf. Softw. Technol. 56 (2), 238–251.
Masud, M.M., L. Khan, and B. Thuraisingham, 2007. A hybrid model to detect malicious executables. In: Proceedings of the IEEE International Conference on Communication (ICC'07), pp. 1443–1448.
Maxion, R.A., 2003. Masquerade detection using enriched command lines. In: Proceedings of International Conference on Dependable Systems and Networks, pp. 5–14.
Mazzariello, C. and F. Oliviero, 2006. An autonomic intrusion detection system based on behavioural network engineering. In: Proceedings of the 25th IEEE International Conference on Computer Communications, INFOCOM 2006. pp. 1–2.
McKinney, S. and D.S. Reeves, 2009. User identification via process profiling: extended abstract. In: Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, pp. 1–4.
Mezghani, M., et al., 2012. A user profile modeling using social annotations: a survey. In: Proceedings of the 21st International Conference Companion on World Wide Web, WWW'12 Companion, pp. 969–976.
Mitchell, R., Chen, I.-R., 2014. A survey of intrusion detection techniques for cyber-physical systems. ACM Comput. Surv. 46 (4), 1–29.
Mitchell, R., Chen, I.-R., 2014. A survey of intrusion detection in wireless network applications. Comput. Commun. 42, 1–23.
Mohammad Faysel, S.H., 2010. Towards cyber defense: research in intrusion detection and intrusion prevention systems. Int. J. Comput. Sci. Netw. Secur. 10 (7), 316–325.
Monowar, H., Bhuyan, D.K.B., Kalita, J.K., 2013. Network anomaly detection: methods, systems and tools. IEEE Commun. Surv. Tutorials 16 (1), 303–336.
Bhaskaran, N., Frank, M., 2011. Lie to me: Deceit detection via online behavioral learning. In: Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011), pp. 24.
Osanaiye, O., Choo, K.-K.R., Dlodloa, M., 2016a. Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework. J. Netw. Comput. Appl. 67, 147–165.
Osanaiye O., Cai H., Choo K.-K. R., Dehghantanha A., Xu Z., Dlodlo M. Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. EURASIP Journal on Wireless Communications and Networking, 2016b, Paper no. 130.
Pannell, G. and H. Ashman, 2010. User modelling for exclusion and anomaly detection: a behavioural intrusion detection system. In: Proceedings of User Modeling, Adaptation, and Personalization, 6075: pp. 207–218.
Pannell, G. and H. Ashman, 2010. Anomaly detection over user profiles for intrusion detection. In: Proceedings of the 8th Australian Information Security Mangement Conference.
Patcha, A., Park, J.-M., 2007. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Comput. Netw. 51 (12), 3448–3470.
Peng, J., Choo, K.-K.R., Ashman, H., 2016. Bit-level n-gram based forensic authorship analysis on social media: Identifying individuals from linguistic profiles. J. Netw. Comput. Appl. 70, 171–182.
Pennington, A., et al., 2010. Storage-based intrusion detection. ACM Trans. Inf. Syst. Secur. 13, 4.
Plagiarism detection, ⟨https://en.wikipedia.org/wiki/Plagiarism_detection⟩.
Quick, D., Martini, B., Choo, K.-K.R., 2013. Cloud storage forensics, 23–61.
Rahman, N.H.A., Choo, K.-K.R., 2015. A survey of information security incident handling in the cloud. Comput. Secur. 49, 45–69.
Ratkiewicz, J., M. Conover, and M. Meiss, 2011a. Detecting and tracking the spread of astroturf memes in microblog streams. Proceedings of the 20th International Conference Companion on World Wide Web, WWW'11; pp. 249–252.
Ratkiewicz, J., et al., 2001b. Detecting and tracking political abuse in social media. In: Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media.
Revett, K., 2009. A bioinformatics based approach to user authentication via keystroke dynamics. Int J Control, Autom Syst 7 (1), 7–15.
Rhee, J., Z. Lin, and D. Xu, 2011. Characterizing kernel malware behavior with kernel data access patterns. In: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, Hong Kong, pp. 207–216.
Rhodes, B., J. Mahaffey, and J. Cannady, 2000. Multiple self-organizing maps for intrusion detection. In: Proceedings of the 23rd National Information Systems Security Conference. Baltimore, MD.
Rodríguez, N.D., et al., 2014. A survey on ontologies for human behavior recognition. ACM Comput. Surv. 46 (4), 1–33.
Shrestha, P., Solorio, T., 2013. Using a Variety of n-Grams for the Detection of Different Kinds of Plagiarism, Notebook for PAN at CLEF 2013. Valencia, España.
Stamatatos, E., 2009. A survey of modern authorship attribution methods. J. Am. Soc. Inf. Sci. Technol. 60 (3), 538–556.
Stamatatos, E., 2009. Intrinsic plagiarism detection using character n-gram profiles. In: Proceedings of the SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 2009), pp. 38–46.
Stein, G., et al., 2005. Decision tree classifier for network intrusiondetection with GA-based feature selection. In: Proceedings of the 43rd Annual Southeast Regional Conference, 2, pp. 136–141.
Stein, B., Lipka, N., Prettenhofer, P., 2010. Intrinsic plagiarism analysis. Lang Resour Eval 45 (1), 63–82.
Tabia, K. and S. Benferhat, 2008. On the use of decision trees as behavioral approaches in intrusion detection. In: Proceedings of the Seventh International Conference on Machine Learning and Applications, pp. 665–670.
Umphress, D., Williams, G., 1985. Identity verification through keyboard characteristics. Int. J. Man-Mach. Stud. 23 (3), 263–273.

Venugopala, D., Hu, G., 2008. Efficient signature based malware detection on mobile devices. Mob. Inf. Syst. 4, 33–49.
Vizer, L.M., Zhou, L., Sears, A., 2009. Automated stress detection using keystroke and linguistic features: an exploratory study. Int. J. Human-Comput. Stud. 67 (10), 870–886.
Wahab, O.A., et al., 2015. A survey on trust and reputation models for web services: single, composite, and communities. Decis. Support. Syst. 74, 121–134.
Woodhams, Jessica; Toye, Kirsty. 2007. An empirical test of the assumptions of case linkage and offender profiling with serial commercial robberies Psychology, Public Policy, and Law, 13, (1): pp. 59–85. ⟨http://dx.doi.org/10.1037/1076-8971.13.1.59⟩.
Wressnegger, C., et al., 2013. A close look on n-grams in intrusion detection: anomaly detection vs. classificatio. In: Proceedings of the 2013 ACM Workshop on Artificial Intelligence and Security, pp. 67–76.
Wu, S.X., Banzhaf, W., 2010. The use of computational intelligence in intrusion detection systems: a review. Appl. Soft Comput. 10 (1), 1–35.
⟨https://www.private-eye.co.uk⟩.
Xie, L., et al., 2010. pBMDS: A Behavior-based Malware Detection System for Cellphone Devices. In: Proceedings of The ACM Conference on Wireless Network Security (WiSec), pp. 37–48.
Yampolskiy, R.V., Govindaraju, V., 2008. Behavioural biometrics: a survey and classification. Int. J. Biom. 1 (1), 81–113.
Yang, Y., 2010. Web user behavioral profiling for user identification. Decis. Support. Syst. 49 (3), 261–271.
Yang, P., Fang, H., 2013. Opin-based User Profile Model Context Suggest, 80–83.
Yang, Y., Padmanabhan, B., 2010. Toward user patterns for online security: Observation time and online user identification. Decis. Support. Syst. 48 (4), 548–558.
Yeung, D.-Y. and Y. Ding, 2002.User profiling for intrusion detection using dynamic and static behavioural models. In: Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Springer-Verlag, pp. 494–505.