Khraisat et al. Cybersecurity (2019) 2:20
https://doi.org/10.1186/s42400-019-0038-7

Cybersecurity

SURVEY Open Access

Survey of intrusion detection systems: techniques, datasets and challenges

Ansam Khraisat*, Iqbal Gondal, Peter Vamplew and Joarder Kamruzzaman

Abstract

Cyber-attacks are becoming more sophisticated and thereby presenting increasing challenges in accurately detecting intrusions. Failure to prevent the intrusions could degrade the credibility of security services, e.g. data confidentiality, integrity, and availability. Numerous intrusion detection methods have been proposed in the literature to tackle computer security threats, which can be broadly classified into Signature-based Intrusion Detection Systems (SIDS) and Anomaly-based Intrusion Detection Systems (AIDS). This survey paper presents a taxonomy of contemporary IDS, a comprehensive review of notable recent works, and an overview of the datasets commonly used for evaluation purposes. It also presents evasion techniques used by attackers to avoid detection and discusses future research challenges to counter such techniques so as to make computer systems more secure.

Keywords: Malware, Intrusion detection system, NSL_KDD, Anomaly detection, Machine learning

Introduction

The evolution of malicious software (malware) poses a critical challenge to the design of intrusion detection systems (IDS). Malicious attacks have become more sophisticated and the foremost challenge is to identify unknown and obfuscated malware, as the malware authors use different evasion techniques for information concealing to prevent detection by an IDS. In addition, there has been an increase in security threats such as zero-day attacks designed to target internet users. Therefore, computer security has become essential as the use of information technology has become part of our daily lives. As a result, various countries such as Australia and the US have been significantly impacted by the zero-day attacks. According to the 2017 Symantec Internet Security Threat Report, more than three billion zero-day attacks were reported in 2016, and the volume and intensity of the zero-day attacks were substantially greater than previously (Symantec, 2017). As highlighted in the Data Breach Statistics in 2017, approximately nine billion data records were lost or stolen by hackers since 2013 (Breach_LeveL_Index, 2017). A Symantec report found that the number of security breach incidents is on the rise. In the past, cybercriminals primarily focused on bank customers, robbing bank accounts or stealing credit cards (Symantec, 2017). However, the new generation of malware has become more ambitious and is targeting the banks themselves, sometimes trying to take millions of dollars in one attack (Symantec, 2017). For that reason, the detection of zero-day attacks has become the highest priority.

High profile incidents of cybercrime have demonstrated the ease with which cyber threats can spread internationally, as a simple compromise can disrupt a business' essential services or facilities. There are a large number of cybercriminals around the world motivated to steal information, illegitimately receive revenues, and find new targets. Malware is intentionally created to compromise computer systems and take advantage of any weakness in intrusion detection systems. In 2017, the Australian Cyber Security Centre (ACSC) critically examined the different levels of sophistication employed by the attackers (Australian, 2017). So there is a need to develop an efficient IDS to detect novel, sophisticated malware. The aim of an IDS is to identify different kinds of malware as early as possible, which cannot be achieved by a traditional firewall. With the increasing volume of computer malware, the development of improved IDSs has become extremely important.

In the last few decades, machine learning has been used to improve intrusion detection, and currently there is a need for an up-to-date, thorough taxonomy and

* Correspondence: a.khraisat@federation.edu.au
Internet Commerce Security Laboratory, Federation University Australia, Mount Helen, Australia

survey of this recent work. There are a large number of related studies using either the KDD-Cup 99 or DARPA 1999 dataset to validate the development of IDSs; however there is no clear answer to the question of which data mining techniques are more effective. Secondly, the time taken for building IDS is not considered in the evaluation of some IDSs techniques, despite being a critical factor for the effectiveness of 'on-line' IDSs.

This paper provides an up to date taxonomy, together with a review of the significant research works on IDSs up to the present time; and a classification of the proposed systems according to the taxonomy. It provides a structured and comprehensive overview of the existing IDSs so that a researcher can become quickly familiar with the key aspects of anomaly detection. This paper also provides a survey of data-mining techniques applied to design intrusion detection systems. The signature-based and anomaly-based methods (i.e., SIDS and AIDS) are described, along with several techniques used in each method. The complexity of different AIDS methods and their evaluation techniques are discussed, followed by a set of suggestions identifying the best methods, depending on the nature of the intrusion. Challenges for the current IDSs are also discussed. Compared to previous survey publications (Patel et al., 2013; Liao et al., 2013a), this paper presents a discussion on IDS dataset problems which are of main concern to the research community in the area of network intrusion detection systems (NIDS). Prior studies such as (Sadotra & Sharma, 2016; Buczak & Guven, 2016) have not completely reviewed IDSs in term of the datasets, challenges and techniques. In this paper, we provide a structured and contemporary, wide-ranging study on intrusion detection system in terms of techniques and datasets; and also highlight challenges of the techniques and then make recommendations.

During the last few years, a number of surveys on intrusion detection have been published. Table 1 shows the IDS techniques and datasets covered by this survey and previous survey papers. The survey on intrusion detection system and taxonomy by Axelsson (Axelsson, 2000) classified intrusion detection systems based on the detection methods. The highly cited survey by Debar et al. (Debar et al., 2000) surveyed detection methods based on the behaviour and knowledge profiles of the attacks. A taxonomy of intrusion systems by Liao et al. (Liao et al., 2013a), has presented a classification of five subclasses with an in-depth perspective on their characteristics: Statistics-based, Pattern-based, Rule-based, State-based and Heuristic-based. On the other hand, our work focuses on the signature detection principle, anomaly detection, taxonomy and datasets.

Existing review articles (e.g., such as (Buczak & Guven, 2016; Axelsson, 2000; Ahmed et al., 2016; Lunt, 1988; Agrawal & Agrawal, 2015)) focus on intrusion detection techniques or dataset issue or type of computer attack and IDS evasion. No articles comprehensively reviewed intrusion detection, dataset problems, evasion techniques, and different kinds of attack altogether. In addition, the development of intrusion-detection systems has been such that several different systems have been proposed in the meantime, and so there is a need for an up-to-date. The updated survey of the taxonomy of intrusion-detection discipline is presented in this paper further enhances taxonomies given in (Liao et al., 2013a; Ahmed et al., 2016).

In view of the discussion on prior surveys, this article focuses on the following:

- Classifying various kinds of IDS with the major types of attacks based on intrusion methods.
- Presenting a classification of network anomaly IDS evaluation metrics and discussion on the importance of the feature selection.
- Evaluation of available IDS datasets discussing the challenges of evasion techniques.

Intrusion detection systems

Intrusion can be defined as any kind of unauthorised activities that cause damage to an information system. This

Table 1 Comparison of this survey and similar surveys: (✓: Topic is covered, ✖ the topic is not covered)

Survey	# of citation (as of 6/1/2019)	Intrusion Detection System Techniques						Dataset issue
		SIDS	AIDS				Hybrid IDS	
			Supervised learning	Unsupervised	Semi-supervised learning	Ensemble methods		
Lunt (1988)	219	✓	✖	✖	✖	✖	✖	✖
Axelsson (2000)	1039	✓	✓	✖	✖	✖	✖	✖
Liao, et al. (2013b)	505	✓	✓	✓	✖	✖	✓	✖
Agrawal and Agrawal (2015)	108	✓	✓	✓	✓	✓	✓	✖
Buczak and Guven (2016)	338	✓	✓	✓	✖	✓	✓	✓
Ahmed, et al. (2016)	181	✖	✓	✓	✖	✖	✖	✓
This survey		✓	✓	✓	✓	✓	✓	✓

means any attack that could pose a possible threat to the information confidentiality, integrity or availability will be considered an intrusion. For example, activities that would make the computer services unresponsive to legitimate users are considered an intrusion. An IDS is a software or hardware system that identifies malicious actions on computer systems in order to allow for system security to be maintained (Liao et al., 2013a). The goal of an IDS is to identify different kinds of malicious network traffic and computer usage, which cannot be identified by a traditional firewall. This is vital to achieving high protection against actions that compromise the availability, integrity, or confidentiality of computer systems. IDS systems can be broadly categorized into two groups: Signature-based Intrusion Detection System (SIDS) and Anomaly-based Intrusion Detection System (AIDS).

Signature-based intrusion detection systems (SIDS)

Signature intrusion detection systems (SIDS) are based on pattern matching techniques to find a known attack; these are also known as Knowledge-based Detection or Misuse Detection (Khraisat et al., 2018). In SIDS, matching methods are used to find a previous intrusion. In other words, when an intrusion signature matches with the signature of a previous intrusion that already exists in the signature database, an alarm signal is triggered. For SIDS, host's logs are inspected to find sequences of commands or actions which have previously been identified as malware. SIDS have also been labelled in the literature as Knowledge-Based Detection or Misuse Detection (Modi et al., 2013).

Figure 1 demonstrates the conceptual working of SIDS approaches. The main idea is to build a database of intrusion signatures and to compare the current set of activities against the existing signatures and raise an alarm if a match is found. For example, a rule in the form of "if: antecedent -then: consequent" may lead to "if (source IP address=destination IP address) then label as an attack ".

SIDS usually gives an excellent detection accuracy for previously known intrusions (Kreibich & Crowcroft, 2004). However, SIDS has difficulty in detecting zero-day attacks for the reason that no matching signature exists in the database until the signature of the new attack is extracted and stored. SIDS are employed in numerous common tools, for instance, Snort (Roesch, 1999) and NetSTAT (Vigna & Kemmerer, 1999).

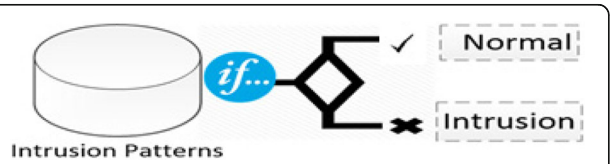

Fig. 1 Conceptual working of SIDS approaches

Traditional approaches to SIDS examine network packets and try matching against a database of signatures. But these techniques are unable to identify attacks that span several packets. As modern malware is more sophisticated it may be necessary to extract signature information over multiple packets. This requires the IDS to recall the contents of earlier packets. With regards to creating a signature for SIDS, generally, there have been a number of methods where signatures are created as state machines (Meiners et al., 2010), formal language string patterns or semantic conditions (Lin et al., 2011).

The increasing rate of zero-day attacks (Symantec, 2017) has rendered SIDS techniques progressively less effective because no prior signature exists for any such attacks. Polymorphic variants of the malware and the rising amount of targeted attacks can further undermine the adequacy of this traditional paradigm. A potential solution to this problem would be to use AIDS techniques, which operate by profiling what is an acceptable behavior rather than what is anomalous, as described in the next section.

Anomaly-based intrusion detection system (AIDS)

AIDS has drawn interest from a lot of scholars due to its capacity to overcome the limitation of SIDS. In AIDS, a normal model of the behavior of a computer system is created using machine learning, statistical-based or knowledge-based methods. Any significant deviation between the observed behavior and the model is regarded as an anomaly, which can be interpreted as an intrusion. The assumption for this group of techniques is that malicious behavior differs from typical user behavior. The behaviors of abnormal users which are dissimilar to standard behaviors are classified as intrusions. Development of AIDS comprises two phases: the training phase and the testing phase. In the training phase, the normal traffic profile is used to learn a model of normal behavior, and then in the testing phase, a new data set is used to establish the system's capacity to generalise to previously unseen intrusions. AIDS can be classified into a number of categories based on the method used for training, for instance, statistical based, knowledge-based and machine learning based (Butun et al., 2014).

The main advantage of AIDS is the ability to identify zero-day attacks due to the fact that recognizing the abnormal user activity does not rely on a signature database (Alazab et al., 2012). AIDS triggers a danger signal when the examined behavior differs from the usual behavior. Furthermore, AIDS has various benefits. First, they have the capability to discover internal malicious activities. If an intruder starts making transactions in a stolen account that are unidentified in the typical user activity, it creates an alarm. Second, it is very difficult for a cybercriminal to recognize what is a normal user

behavior without producing an alert as the system is constructed from customized profiles.

Table 2 presents the differences between signature-based detection and anomaly-based detection. SIDS can only identify well-known intrusions whereas AIDS can detect zero-day attacks. However, AIDS can result in a high false positive rate because anomalies may just be new normal activities rather than genuine intrusions.

Since there is a lack of a taxonomy for anomaly-based intrusion detection systems, we have identified five subclasses based on their features: Statistics-based, Pattern-based, Rule-based, State-based and Heuristic-based as shown in Table 3.

Intrusion data sources

The previous two sections categorised IDS on the basis of the methods used to identify intrusions. IDS can also be classified based on the input data sources used to detect abnormal activities. In terms of data sources, there are generally two types of IDS technologies, namely Host-based IDS (HIDS) and Network-based IDS (NIDS). HIDS inspect data that originates from the host system and audit sources, such as operating system, window server logs, firewalls logs, application system audits, or database logs. HIDS can detect insider attacks that do not involve network traffic (Creech & Hu, 2014a).

NIDS monitors the network traffic that is extracted from a network through packet capture, NetFlow, and other network data sources. Network-based IDS can be used to monitor many computers that are joined to a network. NIDS is able to monitor the external malicious activities that could be initiated from an external threat at an earlier phase, before the threats spread to another computer system. On the other hand, NIDSs have limited ability to inspect all data in a high bandwidth network because of the volume of data passing through modern high-speed communication networks (Bhuyan et al., 2014). NIDS deployed at a number of positions within a particular network topology, together with HIDS and firewalls, can provide a concrete, resilient, and multi-tier protection against both external and insider attacks.

Table 4 shows a summary of comparisons between HIDS and NIDS.

Creech et al. proposed a HIDS methodology applying discontinuous system call patterns, with the aim to raise detection rates while decreasing false alarm rates (Creech, 2014). The main idea is to use a semantic structure to kernel level system calls to understand anomalous program behaviour.

As shown in Table 5 a number of AIDS systems have also been applied in Network Intrusion Detection System (NIDS) and Host Intrusion Detection System (HIDS) to increase the detection performance with the use of machine learning, knowledge-based and statistical schemes. Table 5 also provides examples of current intrusion detection approaches, where types of attacks are presented in the detection capability field. Data source comprises system calls, application programme interfaces, log files, data packets obtained from well-known attacks. These data source can be beneficial to classify intrusion behaviors from abnormal actions.

Techniques for implementing AIDS

This section presents an overview of AIDS approaches proposed in recent years for improving detection accuracy and reducing false alarms.

AIDS methods can be categorized into three main groups: Statistics-based (Chao et al., 2015), knowledge-based (Elhag et al., 2015; Can & Sahingoz, 2015), and machine learning-based (Buczak & Guven, 2016; Meshram & Haas, 2017). The statistics-based approach involves collecting and examining every data record in a set of items and building a statistical model of normal user behavior. On the other hand, knowledge-based tries to identify the requested actions from existing system data such as protocol specifications and network traffic instances, while machine-learning methods acquire complex pattern-matching capabilities from training data.

Table 2 Comparisons of intrusion detection methodologies

		Advantages	Disadvantages
Detection methods	SIDS	• Very effective in identifying intrusions with minimum false alarms (FA). • Promptly identifies the intrusions. • Superior for detecting the known attacks. • Simple design	• Needs to be updated frequently with a new signature. • SIDS is designed to detect attacks for known signatures. When a previous intrusion has been altered slightly to a new variant, then the system would be unable to identify this new deviation of the similar attack. • Unable to detect the zero-day attack. • Not suitable for detecting multi-step attacks. • Little understanding of the insight of the attacks
	AIDS	• Could be used to detect new attacks. • Could be used to create intrusion signature	• AIDS cannot handle encrypted packets, so the attack can stay undetected and can present a threat. • High false positive alarms. • Hard to build a normal profile for a very dynamic computer system. • Unclassified alerts. • Needs initial training.

Table 3 Detection methodology characteristics for intrusion-detection systems

Detection Methodology	Examples	Characteristics
Statistics based: analyzes the network traffic using complex statistical algorithms to process the information.	Bhuyan, et al. (2014)	•Needs a large amount of knowledge of statistics •Simple but less accurate •Real-time
Pattern-based: identifies the characters, forms, and patterns in the data.	Liao, et al. (2013a) Riesen and Bunke (2008)	•Easy to implement •Hash function could be used for identification.
Rule-based: uses an attack "signature" to detect a potential attack on the suspicious network traffic.	Hall, et al. (2009)	•The computational cost of rule-based systems could be very high because rules need pattern matching. •It is very hard to estimate what actions are going to occur and when •Requires a large number of rules for determining all possible attacks. •Low false positive rate •High detection rate
State-based: examines a stream of events to identify any possible attack.	Kenkre, et al. (2015a)	•Probabilistic, self-training •Low false positive rate.
Heuristic-based: identifies any abnormal activity that is out of the ordinary activity.	Abbasi, et al. (2014) Butun, et al. (2014)	•It needs knowledge and experience •Experimental and evolutionary learning

These three classes along with examples of their subclasses are shown in Fig. 2.

Statistics-based techniques

A statistics-based IDS builds a distribution model for normal behaviour profile, then detects low probability events and flags them as potential intrusions. Statistical AIDS essentially takes into account the statistical metrics such as the median, mean, mode and standard deviation of packets. In other words, rather than inspecting data traffic, each packet is monitored, which signifies the fingerprint of the flow. Statistical AIDS are employed to identify any type of differences in the present behavior from normal behavior. Statistical IDS normally use one of the following models.

Univariate: "Uni" means "one", so it means the data has only one variable. This technique is used when a statistical normal profile is created for only one measure of behaviours in computer systems. Univariate IDS look for abnormalities in each individual metric (Ye et al., 2002).

Multivariate: It is based on relationships among two or more measures in order to understand the relationships between variables. This model would be valuable if experimental data show that better classification can be achieved from combinations of correlated measures rather than analysing them separately. Ye et al. examine a multivariate quality control method to identify intrusions by building a long-term profile of normal activities (Ye et al., 2002). The main challenge for multivariate statistical IDs is that it is difficult to estimate distributions for high-dimensional data.

Time series model: A time series is a series of observations made over a certain time interval. A new observation

Table 4 Comparison of IDS technology types based on their positioning within the computer system

		Advantages	Disadvantages	Data source
Technology	HIDS	• HIDS can check end-to-end encrypted communications behaviour. • No extra hardware required. • Detects intrusions by checking hosts file system, system calls or network events. • Every packet is reassembled • Looks at the entire item, not streams only	• Delays in reporting attacks • Consumes host resources • Needs to be installed on each host. • It can monitor attacks only on the machine where it is installed.	• Audits records, log files, Application Program Interface (API), rule patterns, system calls.
	NIDS	•Detects attacks by checking network packets. •Not required to install on each host. •Can check various hosts at the same period. •Capable of detecting the broadest ranges of network protocols	•Challenge is to identify attacks from encrypted traffic. •Dedicated hardware is required. •It supports only identification of network attacks. •Difficult to analysis high-speed network. •The most serious threat is the insider attack.	•Simple Network Management Protocol (SNMP) •Network packets (TCP/UDP/ICMP), •Management Information Base (MIB) •Router NetFlow records

Table 5 Comparisons of IDS technology types, using examples from the literature. "P" indicates pre-defined attacks and "Z" indicates zero-day attacks

Detection Source			HIDS	NIDS	Capability
Detection methods	SIDS		Wagner and Soto (2002)	Hubballi and Suryanarayanan (2014)	P
	AIDS	Statistics based	Ara, Louzada & Diniz (2017)	Tan, et al. (2014); Camacho, et al. (2016)	Z
		Knowledge-based	Mitchell and Chen (2015) Creech and Hu (2014b)	Hendry and Yang (2008) Shakshuki, et al. (2013) Zargar, et al. (2013)	
		Machine learning	Du, et al. (2014) Wang, et al. (2010)	Elhag, et al. (2015); Kim, et al. (2014); Hu, et al. (2014)	
	SIDS+ AIDS		Alazab, et al. (2014); Stavroulakis and Stamp (2010); Liu, et al. (2015)		P + Z

is abnormal if its probability of occurring at that time is too low. Viinikka et al. used time series for processing intrusion detection alert aggregates (Viinikka et al., 2009). Qingtao et al. presented a method for detecting network abnormalities by examining the abrupt variation found in time series data (Qingtao & Zhiqing, 2005). The feasibility of this technique was validated through simulated experiments.

Knowledge-based techniques

This group of techniques is also referred toas an expert system method. This approach requires creating a knowledge base which reflects the legitimate traffic profile. Actions which differ from this standard profile are treated as an intrusion. Unlike the other classes of AIDS, the standard profile model is normally created based on human knowledge, in terms of a set of rules that try to define normal system activity.

The main benefit of knowledge-based techniques is the capability to reduce false-positive alarms since the system has knowledge about all the normal behaviors. However, in a dynamically changing computing environment, this kind of IDS needs a regular update on knowledge for the expected normal behavior which is a time-consuming task as gathering information about all normal behaviors is very difficult.

Finite state machine (FSM): FSM is a computation model used to represent and control execution flow.

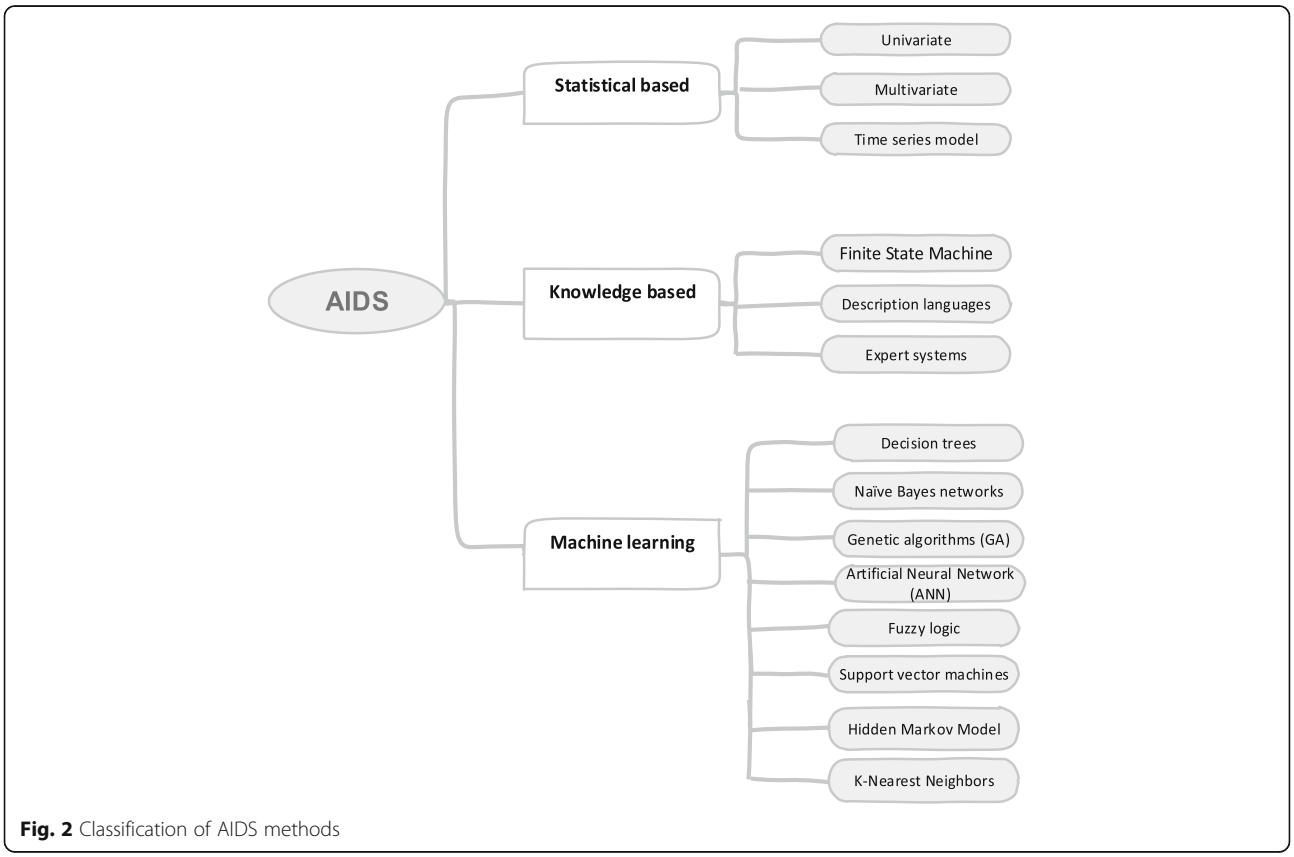

Fig. 2 Classification of AIDS methods

This model could be applied in intrusion detection to produce an intrusion detection system model. Typically, the model is represented in the form of states, transitions, and activities. A state checks the history data. For instance, any variations in the input are noted and based on the detected variation transition happens (Walkinshaw et al., 2016). An FSM can represent legitimate system behaviour, and any observed deviation from this FSM is regarded as an attack.

Description Language: Description language defines the syntax of rules which can be used to specify the characteristics of a defined attack. Rules could be built by description languages such as N-grammars and UML (Studnia et al., 2018).

Expert System: An expert system comprises a number of rules that define attacks. In an expert system, the rules are usually manually defined by a knowledge engineer working in collaboration with a domain expert (Kim et al., 2014).

Signature analysis: it is the earliest technique applied in IDS. It relies on the simple idea of string matching. In string matching, an incoming packet is inspected, word by word, with a distinct signature. If a signature is matched, an alert is raised. If not, the information in the traffic is then matched to the following signature on the signature database (Kenkre et al., 2015b).

AIDS based on machine learning techniques

Machine learning is the process of extracting knowledge from large quantities of data. Machine learning models comprise of a set of rules, methods, or complex "transfer functions" that can be applied to find interesting data patterns, or to recognise or predict behaviour (Dua & Du, 2016).

Machine learning techniques have been applied extensively in the area of AIDS. Several algorithms and techniques such as clustering, neural networks, association rules, decision trees, genetic algorithms, and nearest neighbour methods, have been applied for discovering the knowledge from intrusion datasets (Kshetri & Voas, 2017; Xiao et al, 2018).

Some prior research has examined the use of different techniques to build AIDSs. Chebrolu et al. examined the performance of two feature selection algorithms involving Bayesian networks (BN) and Classification Regression Trees (CRC) and combined these methods for higher accuracy (Chebrolu et al., 2005).

Bajaj et al. proposed a technique for feature selection using a combination of feature selection algorithms such as Information Gain (IG) and Correlation Attribute evaluation. They tested the performance of the selected features by applying different classification algorithms such as C4.5, naïve Bayes, NB-Tree and Multi-Layer Perceptron (Khraisat et al., 2018; Bajaj & Arora, 2013). A genetic-fuzzy rule mining method has been used to evaluate the importance of IDS features (Elhag et al., 2015). Thaseen et al. proposed NIDS by using Random Tree model to improve the accuracy and reduce the false alarm rate (Thaseen & Kumar, 2013). Subramanian et al. proposed classifying NSL-KDD dataset using decision tree algorithms to construct a model with respect to their metric data and studying the performance of decision tree algorithms (Subramanian et al., 2012).

Various AIDSs have been created based on machine learning techniques as shown in Fig. 3. The objective of using machine learning techniques is to create IDS with improved accuracy and less requirement for human knowledge. In the last few years, the quantity of AIDS which have used machine learning methods has been increasing. A key focus of IDS based on machine learning research is to detect patterns and build intrusion detection system based on the dataset. Generally, there are two kinds of machine learning methods, supervised and unsupervised.

Supervised learning in intrusion detection system

This section presents various supervised learning techniques for IDS. Each technique is presented in detail, and references to important research publications are presented.

Supervised learning-based IDS techniques detect intrusions by using labeled training data. A supervised learning approach usually consists of two stages, namely training and testing. In the training stage, relevant features and classes are identified and then the algorithm learns from these data samples. In supervised learning IDS, each record is a pair, containing a network or host data source and an associated output value (i.e., label), namely intrusion or normal. Next, feature selection can be applied for eliminating unnecessary features. Using the training data for selected features, a supervised learning technique is then used to train a classifier to learn the inherent relationship that exists between the input data and the labelled output value. A wide variety of supervised learning techniques have been explored in the literature, each with its advantages and disadvantages. In the testing stage, the trained model is used to classify the unknown data into intrusion or normal class. The resultant classifier then becomes a model which, given a set of feature values, predicts the class to which the input data might belong. Figure 4 shows a general approach for applying classification techniques. The performance of a classifier in its ability to predict the correct class is measured in terms of a number of metrics is discussed in Section 4.

There are many classification methods such as decision trees, rule-based systems, neural networks, support vector machines, naïve Bayes and nearest-neighbor. Each technique uses a learning method to build a classification model. However, a suitable classification approach

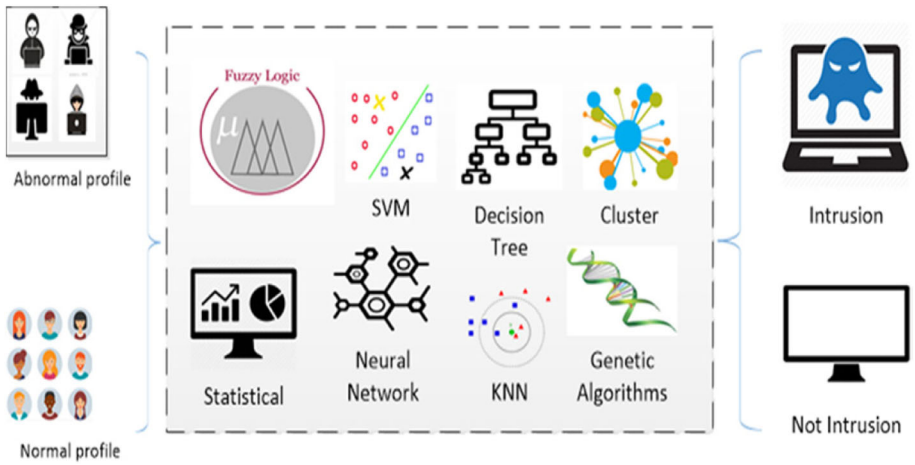

Fig. 3 Conceptual working of AIDS approaches based on machine learning

should not only handle the training data, but it should also identify accurately the class of records it has not ever seen before. Creating classification models with reliable generalization ability is an important task of the learning algorithm.

Decision trees: A decision tree comprises of three basic components. The first component is a decision node, which is used to identify a test attribute. The second is a branch, where each branch represents a possible decision based on the value of the test attribute. The third is a leaf that comprises the class to which the instance belongs (Rutkowski et al., 2014). There are many different decision trees algorithms including ID3 (Quinlan, 1986), C4.5 (Quinlan, 2014) and CART (Breiman, 1996).

Naïve Bayes: This approach is based on applying Bayes' principle with robust independence assumptions among the attributes. Naïve Bayes answers questions such as "what is the probability that a particular kind of attack is occurring, given the observed system activities?" by applying conditional probability formulae. Naïve Bayes relies on the features that have different probabilities of occurring in attacks and in normal behavior. Naïve Bayes classification model is one of the most prevalent models in IDS due to its ease of use and calculation efficiency, both of which are taken from its conditional independence assumption property (Yang & Tian, 2012). However, the system does not operate well if this independence assumption is not valid, as was demonstrated on the KDD'99 intrusion detection dataset which has complex attribute dependencies (Koc et al., 2012). The results also reveal that the Naïve Bayes model has reduced accuracy for large datasets. A further study showed that the more sophisticated Hidden Naïve Bayes (HNB) model can be applied to IDS tasks that involve high dimensionality, extremely interrelated attributes and high-speed networks (Koc et al., 2012).

Genetic algorithms (GA): Genetic algorithms are a heuristic approach to optimization, based on the principles of evolution. Each possible solution is represented as a series of bits (genes) or chromosome, and the quality of the solutions improves over time by the application of selection and reproduction operators, biased to favour fitter solutions. In applying a genetic algorithm to the intrusion classification problem, there are typically two types of chromosome encoding: one is according to clustering to generate binary chromosome coding method; another is specifying the cluster center (clustering prototype matrix) by an integer coding chromosome.

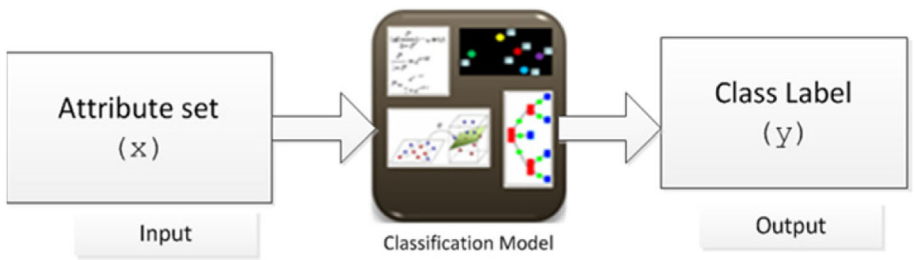

Fig. 4 Classification as the task

Murray et al., has used GA to evolve simple rules for network traffic (Murray et al., 2014). Every rule is represented by a genome and the primary population of genomes is a number of random rules. Each genome is comprised of different genes which correspond to characteristics such as IP source, IP destination, port source, port destination and 1 protocol type (Hoque & Bikas, 2012).

Artificial Neural Network (ANN): ANN is one of the most broadly applied machine-learning methods and has been shown to be successful in detecting different malware. The most frequent learning technique employed for supervised learning is backpropagation (BP) algorithm. The BP algorithm assesses the gradient of the network's error with respect to its modifiable weights. However, for ANN-based IDS, detection precision, particularly for less frequent attacks, and detection accuracy still need to be improved. The training dataset for less-frequent attacks is small compared to that of more-frequent attacks and this makes it difficult for the ANN to learn the properties of these attacks correctly. As a result, detection accuracy is lower for less frequent attacks. In the information security area, huge damage can occur if low-frequency attacks are not detected. For instance, if the User to Root (U2R) attacks evade detection, a cybercriminal can gain the authorization privileges of the root user and thereby carry out malicious activities on the victim's computer systems. In addition the less common attacks are often outliers (Wang et al., 2010). ANNs often suffer from local minima and thus learning can become very time-consuming. The strength of ANN is that, with one or more hidden layers, it is able to produce highly nonlinear models which capture complex relationships between input attributes and classification labels. With the development of many variants such as recurrent and convolutional NNs, ANNs are powerful tools in many classification tasks including IDS.

Fuzzy logic: This technique is based on the degrees of uncertainty rather than the typical true or false Boolean logic on which the contemporary PCs are created. Therefore, it presents a straightforward way of arriving at a final conclusion based upon unclear, ambiguous, noisy, inaccurate or missing input data. With a fuzzy domain, fuzzy logic permits an instance to belong, possibly partially, to multiple classes at the same time. Therefore, fuzzy logic is a good classifier for IDS problems as the security itself includes vagueness, and the borderline between the normal and abnormal states is not well identified. In addition, the intrusion detection problem contains various numeric features in the collected data and several derived statistical metrics. Building IDSs based on numeric data with hard thresholds produces high false alarms. An activity that deviates only slightly from a model could not be recognized or a minor change in normal activity could produce false alarms. With fuzzy logic, it is possible to model this minor abnormality to keep the false rates low. Elhag et al. showed that with fuzzy logic, the false alarm rate in determining intrusive actions could be decreased. They outlined a group of fuzzy rules to describe the normal and abnormal activities in a computer system, and a fuzzy inference engine to define intrusions (Elhag et al., 2015).

Support Vector Machines (SVM): SVM is a discriminative classifier defined by a splitting hyperplane. SVMs use a kernel function to map the training data into a higher-dimensioned space so that intrusion is linearly classified. SVMs are well known for their generalization capability and are mainly valuable when the number of attributes is large and the number of data points is small. Different types of separating hyperplanes can be achieved by applying a kernel, such as linear, polynomial, Gaussian Radial Basis Function (RBF), or hyperbolic tangent. In IDS datasets, many features are redundant or less influential in separating data points into correct classes. Therefore, features selection should be considered during SVM training. SVM can also be used for classification into multiple classes. In the work by Li et al., an SVM classifier with an RBF kernel was applied to classify the KDD 1999 dataset into predefined classes (Li et al., 2012). From a total of 41 attributes, a subset of features was carefully chosen by using feature selection method.

Hidden Markov Model (HMM): HMM is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unseen data. Prior research has shown that HMM analysis can be applied to identify particular kinds of malware (Annachhatre et al., 2015). In this technique, a Hidden Markov Model is trained against known malware features (e.g., operation code sequence) and once the training stage is completed, the trained model is applied to score the incoming traffic. The score is then contrasted to a predefined threshold, and a score greater than the threshold indicates malware. Likewise, if the score is less than the threshold, the traffic is identified as normal.

K-Nearest Neighbors (KNN) classifier: The k-Nearest Neighbor (k-NN) techniques is a typical non-parametric classifier applied in machine learning (Lin et al., 2015). The idea of these techniques is to name an unlabelled data sample to the class of its k nearest neighbors (where k is an integer defining the number of neighbours to be considered). Figure 5 illustrates a K-Nearest Neighbors classifier where k = 5. The point X represents an instance of unlabelled date which needs to be classified. Amongst the five nearest neighbours of X there are three similar patterns from the class Intrusion and two from the class Normal. Taking a majority vote enables the assignment of X to the Intrusion class.

k-NN can be appropriately applied as a benchmark for all the other classifiers because it provides a good classification performance in most IDSs (Lin et al., 2015).

Unsupervised learning in intrusion detection system

Unsupervised learning is a form of machine learning technique used to obtain interesting information from input datasets without class labels. The input data points are normally treated as a set of random variables. A joint density model is then created for the data set. In supervised learning, the output labels are given and used to train the machine to get the required results for an unseen data point, while in unsupervised learning, no labels are given, and instead the data is grouped automatically into various classes through the learning process. In the context of developing an IDS, unsupervised learning means, use of a mechanism to identify intrusions by using unlabelled data to a train the model.

As shown in Fig. 6, once records are clustered, all of the cases that appear in small clusters are labelled as an intrusion because the normal occurrences should produce sizable clusters compared to the anomalies. In addition, malicious intrusions and normal instances are dissimilar, thus they do not fall into the identical cluster.

K-means: The K-means techniques is one of the most prevalent techniques of clustering analysis that aims to separate 'n' data objects into 'k' clusters in which each data object is selected in the cluster with the nearest mean. It is a distance-based clustering technique and it does not need to compute the distances between all combinations of records. It applies a Euclidean metric as a similarity measure. The number of clusters is determined by the user in advance. Typically several solutions will be tested before accepting the most appropriate one. Annachhatre et.al. used the K-means clustering algorithm to identify different host behaviour profiles (Annachhatre et al., 2015). They have proposed new distance metrics which can be used in the k-means algorithm to closely relate the clusters. They have clustered data into several clusters and associated them with known behavior for evaluation. Their outcomes have revealed that k-means clustering is a better approach to classify the data using unsupervised methods for intrusion detection when several kinds of datasets are available. Clustering could be used in IDS for reducing intrusion signatures, generate a high-quality signature or group similar intrusion.

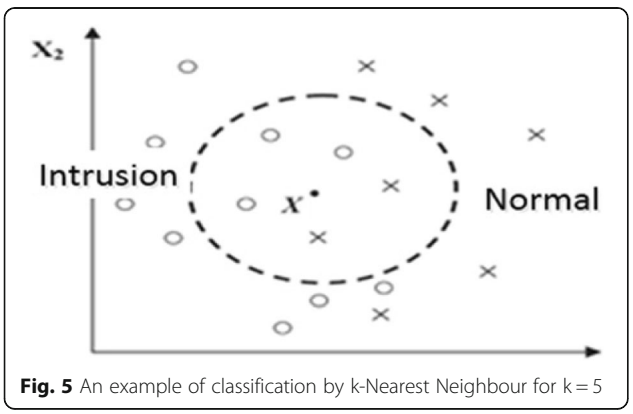

Fig. 5 An example of classification by k-Nearest Neighbour for k = 5

Hierarchical Clustering: This is a clustering technique which aims to create a hierarchy of clusters. Approaches for hierarchical clustering are normally classified into two categories:

(i) Agglomerative- bottom-up clustering techniques where clusters have sub-clusters, which in turn have sub-clusters and pairs of clusters are combined as one moves up the hierarchy.
(ii) Divisive - hierarchical clustering algorithms where iteratively the cluster with the largest diameter in feature space is selected and separated into binary sub-clusters with lower range.

A lot of work has been done in the area of the cyber-physical control system (CPCS) with attack detection and reactive attack mitigation by using unsupervised learning. For example, a redundancy-based resilience approach was proposed by Alcara (Alcaraz, 2018). He proposed a dedicated network sublayer that has the capability to handle the context by regularly collecting consensual information from the driver nodes controlled in the control network itself, and discriminating view differences through data mining techniques such as k-means and k-nearest neighbour. Chao Shen et al. proposed Hybrid-Augmented device fingerprinting for IDS in Industrial Control System Networks. They used different machine learning techniques to analyse network packets to filter anomaly traffic to detect in the intrusions in ICS networks (Shen et al., 2018).

Semi-supervised learning

Semi-supervised learning falls between supervised learning (with totally labelled training data) and unsupervised learning (without any categorized training data). Researchers have shown that semi-supervised learning could be used in conjunction with a small amount of labelled data classifier's performance for the IDSs with less time and costs needed. This is valuable as for many IDS issues, labelled data can be rare or occasional (Ashfaq et al., 2017).

A number of different techniques for semi-supervised learning have been proposed, such as the Expectation Maximization (EM) based algorithms (Goldstein, 2012), self-training (Blount et al., 2011; Lyngdoh et al., 2018), co-training (Rath et al., 2017), Semi-Supervised SVM (Ashfaq et al., 2017), graph-based methods (Sadreazami

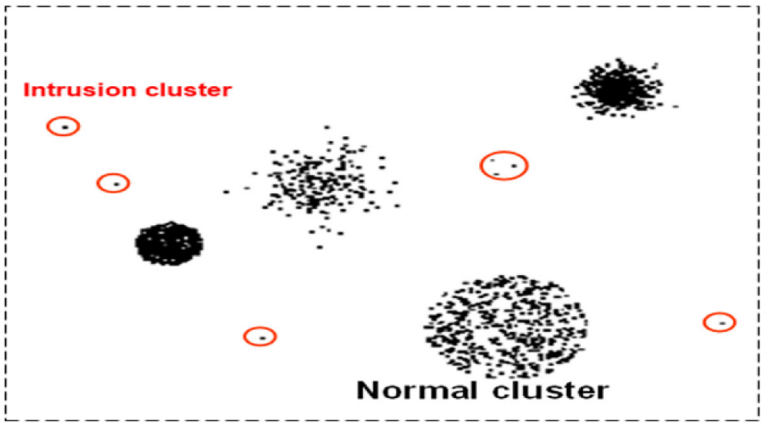

Fig. 6 Using Clustering for Intrusion Detection

et al., 2018), and boosting based semi-supervised learning methods (Yuan et al., 2016).

Rana et al. propose a novel fuzzy-based semi-supervised learning approach by applying unlabelled samples aided with a supervised learning algorithm to enhance the classifier's performance for the IDSs. A single hidden layer feed-forward neural network (SLFN) is trained to output a fuzzy membership vector, and the sample categorization (low, mid, and high fuzziness categories) on unlabelled samples is performed using the fuzzy quantity (Ashfaq et al., 2017). The classifier is retrained after incorporating each category separately into the original training set. Their experimental results using this semi-supervised of intrusion detection on the NSL-KDD dataset show that unlabelled samples belonging to low and high fuzziness groups cause foremost contributions to enhance the accuracy of IDS contrasted to traditional.

Ensemble methods

Multiple machine learning algorithms can be used to obtain better predictive performance than any of the constituent learning algorithms alone. A number of different ensemble methods have been proposed, such as Boosting, Bagging and Stacking.

Boosting refers to a family of algorithms that are able to transform weak learners to strong learners. Bagging means training the same classifier on different subsets of same dataset. Stacking combines various classification via a meta-classifier (Aburomman & Reaz, 2016). The base level models are built based on a whole training set, then the meta-model is trained on the outputs of the base level model as attributes.

Jabbar et al. proposed an ensemble classifier which is built using Random Forest and also the Average One-Dependence Estimator (AODE which solves the attribute dependency problem in Naïve Bayes classifier. Random Forest (RF) enhances precision and reduces false alarms (Jabbar et al., 2017). Combining both approaches in an ensemble results in improved accuracy over either technique applied independently.

Hybrid based techniques

Traditional IDSs have limitations: that they cannot be easily modified, inability to identify new malicious attacks, low accuracy and high false alarms. Where AIDS has a limitation such as high false positive rate. Hybrid IDS is based on the combination of SIDS and AIDS. A Hybrid IDS overcomes the disadvantage of SIDS and AIDS. Farid et al. (Farid et al., 2010) proposed hybrid IDS by using Naive Bayes and decision tree based and achieved detection rate of 99.63% on the KDD'99 dataset.

Performance metrics for IDS

There are many classification metrics for IDS, some of which are known by multiple names. Table 6 shows the confusion matrix for a two-class classifier which can be used for evaluating the performance of an IDS. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

IDS are typically evaluated based on the following standard performance measures:

- True Positive Rate (TPR): It is calculated as the ratio between the number of correctly predicted attacks and the total number of attacks. If all intrusions are

Table 6 Confusion Matrix for IDS System

Actual Class	Predicted Class		
	Class	Normal	Attack
	Normal	True negative (TN)	False Positive (FP)
	Attack	False Negative (FN)	True positive (TP)

detected then the TPR is 1 which is extremely rare for an IDS. TPR is also called a Detection Rate (DR) or the Sensitivity. The TPR can be expressed mathematically as

$$TPR = \frac{TP}{TP + FN}$$

- False Positive Rate (FPR): It is calculated as the ratio between the number of normal instances incorrectly classified as an attack and the total number of normal instances.

$$FPR = \frac{FP}{FP + TN}$$

- False Negative Rate (FNR): False negative means when a detector fails to identify an anomaly and classifies it as normal. The FNR can be expressed mathematically as:

$$FNR = \frac{FN}{FN + TP}$$

- Classification rate (CR) or Accuracy: The CR measures how accurate the IDS is in detecting normal or anomalous traffic behavior. It is described as the percentage of all those correctly predicted instances to all instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Receiver Operating Characteristic (ROC) curve: ROC has FPR on the x-axis and TPR on the y-axis. In ROC curve the TPR is plotted as a function of the FPR for different cut-off points. Each point on the ROC curve represents a FPR and TPR pair corresponding to a certain decision threshold. As the threshold for classification is varied, a different point on the ROC is selected with different False Alarm Rate (FAR) and different TPR. A test with perfect discrimination (no overlap in the two distributions) has a ROC curve that passes through the upper left corner (100% sensitivity, 100% specificity). The ROC Curve is shown in Fig. 7.

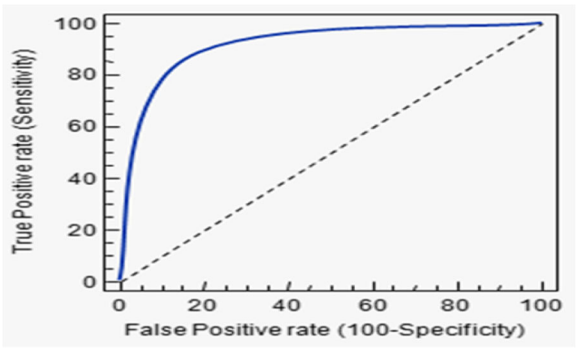

Fig. 7 ROC curve

Intrusion detection datasets

The evaluation datasets play a vital role in the validation of any IDS approach, by allowing us to assess the proposed method's capability in detecting intrusive behavior. The datasets used for network packet analysis in commercial products are not easily available due to privacy issues. However, there are a few publicly available datasets such as DARPA, KDD, NSL-KDD and ADFA-LD and they are widely used as benchmarks. Existing datasets that are used for building and comparative evaluation of IDS are discussed in this section along with their features and limitations.

DARPA / KDD Cup99

The earliest effort to create an IDS dataset was made by DARPA (Defence Advanced Research Project Agency) in 1998 and they created the KDD98 (Knowledge Discovery and Data Mining (KDD)) dataset. In 1998, DARPA introduced a programme at the MIT Lincoln Labs to provide a comprehensive and realistic IDS benchmarking environment (MIT Lincoln Laboratory, 1999). Although this dataset was an important contribution to the research on IDS, its accuracy and capability to consider real-life conditions have been widely criticized (Creech & Hu, 2014b).

These datasets were collected using multiple computers connected to the Internet to model a small US Air Force base of restricted personnel. Network packets and host log files were collected. Lincoln Labs built an experimental testbed to obtain 2 months of TCP packets dump for a Local Area Network (LAN), modelling a usual US Air Force LAN. They modelled the LAN as if it were a true Air Force environment, but interlaced it with several simulated intrusions.

The collected network packets were around four gigabytes containing about 4,900,000 records. The test data of 2 weeks had around 2 million connection records, each of which had 41 features and was categorized as normal or abnormal.

The extracted data is a series of TCP sessions starting and ending at well-defined times, between which data flows to and from a source IP address to a target IP address, which contains a large variety of attacks simulated in a military network environment. The 1998 DARPA Dataset was used as the basis to derive the KDD Cup99 dataset which has been used in Third International Knowledge Discovery and Data Mining Tools Competition (KDD, 1999). The 41 features of the KDD Cup99 dataset are presented in Table 7.

These datasets are out-of-date as they do not contain records of recent malware attacks. For example, attackers' behaviors are different in different network topologies, operating systems, and software and crime toolkits. Nevertheless, KDD99 remains in use as a benchmark within IDS research community and is still presently being used by researchers (Alazab et al., 2014; Duque & Omar, 2015; Ji et al., 2016).

CAIDA

This dataset contains network traffic traces from Distributed Denial-of-Service (DDoS) attacks, and was collected in 2007 (Hick et al., 2007). This type of denial-of-service attack attempts to interrupt normal traffic of a targeted computer, or network by overwhelming the target with a flood of network packets, preventing regular traffic from reaching its legitimate destination computer. One disadvantage of the CAIDA dataset is that it does not contain a diversity of the attacks. In addition, the gathered data does not contain features from the whole network which makes it difficult to distinguish between abnormal and normal traffic flows.

NSL-KDD

NSL-KDD is a public dataset, which has been developed from the earlier KDD cup99 dataset (Tavallaee et al., 2009). A statistical analysis performed on the cup99 dataset raised important issues which heavily influence the intrusion detection accuracy, and results in a misleading evaluation of AIDS (Tavallaee et al., 2009).

The main problem in the KDD data set is the huge amount of duplicate packets. Tavallaee et al. analyzed KDD training and test sets and revealed that approximately 78% and 75% of the network packets are duplicated in both the training and testing dataset (Tavallaee et al., 2009). This huge quantity of duplicate instances in the training set would influence machine-learning methods to be biased towards normal instances and thus prevent them from learning irregular instances which are typically more damaging to the computer system. Tavallaee et al. built the NSL-KDD dataset in 2009 from the KDD Cup'99 dataset to resolve the matters stated above by eliminating duplicated records (Tavallaee et al., 2009). The NSL-KDD train dataset consists of 125,973 records and the test dataset contains 22,544 records. The size of the NSL-KDD dataset is sufficient to make it practical to use the whole NSL-KDD dataset without the necessity to sample randomly. This has produced consistent and comparable results from various research works. The NSL_KDD dataset comprises 22 training intrusion attacks and 41 attributes (i.e., features). In this dataset, 21 attributes refer to the connection itself and 19 attributes describe the nature of connections within the same host (Tavallaee et al., 2009).

Table 7 The 41 features of KDD Cup99 dataset

Label	Network data feature	Label	Network data feature	Label	Network data feature	Label	Network data feature
A	duration	L	Logged in	W	count	AH	dst_host_same_srv_rate
B	protocol-type	M	num_ comprised	X	srv_count	AI	dst_host_diff_srv_rate
C	service	N	root_shell	Y	serror_rate	AJ	dst_host_same_src_port_rate
D	flag	O	Stu attempted	Z	srv_serror_rate	AK	dst_host_srv_diff_host_rate
E	src_bytes	P	num_root	AA	rerror_rate	AL	dst_host_serror_rate
F	dst_bytes	Q	Num of file	AB	srv_rerror_rate	AM	dst_host_srv_serror_rate
G	land	R	Number of shell	AC	same_srv_rate	AN	dst_host_rerror_rate
H	wrong_fragment	S	num_access_files	AD	diff_srv_rate	AO	dst_host_srv_rerror_rate
I	urgent	T	num_outbound_cmds	AE	srv_diff_host_rate		
J	hot	U	Is host login	AF	dst_host_count		
K	num_falied_logins	V	Is guest login	AG	dst_host_srv_count		

ISCX 2012

In this dataset, real network traffic traces were analyzed to identify normal behaviour for computers from real traffic of HTTP, SMTP, SSH, IMAP, POP3, and FTP protocols (Shiravi et al., 2012). This dataset is based on realistic network traffic, which is labeled and contains diverse attacks scenarios.

ADFA-LD and ADFA-WD

Researchers at the Australian Defence Force Academy created two datasets (ADFA-LD and ADFA-WD) as public datasets that represent the structure and methodology of the modern attacks (Creech, 2014). The datasets contain records from both Linux and Windows operating systems; they are created from the evaluation of system-call-based HIDS. Ubuntu Linux version 11.04 was used as the host operating system to build ADFA-LD (Creech & Hu, 2014b). Some of the attack instances in ADFA-LD were derived from new zero-day malware, making this dataset suitable for highlighting differences between SIDS and AIDS approaches to intrusion detection. It comprises three dissimilar data categories, each group of data containing raw system call traces. Each training dataset was gathered from the host for normal activities, with user behaviors ranging from web browsing to LATEX document preparation. Table 8 shows some of the ADFA-LD features with the type and the description for each feature.

ADFA-LD also incorporates system call traces of different types of attacks. The ADFA Windows Dataset (ADFA-WD) provides a contemporary Windows dataset for evaluation of HIDS. Table 9 shows the number of systems calls for each category of AFDA-LD and AFDA-WD Table 10 describes details of each attack class in the ADFA-LD dataset. Table 11 lists the ADFA-WD Vectors and Effects.

Table 8 Features of ADFA-LD dataset (Creech, 2014)

Name	Type	Description
srcip	nominal	Source IP address
sport	integer	Source port number
dstip	nominal	Destination IP address
dsport	integer	Destination port number
proto	nominal	Transaction protocol
state	nominal	Indicates to the state and its dependent protocol
dur	Float	Record total duration
sbytes	Integer	Source to destination transaction bytes
dbytes	Integer	Destination to source transaction bytes
sttl	Integer	Source to destination time to live value
dttl	Integer	Destination to source time to live value
sloss	Integer	Source packets retransmitted or dropped
dloss	Integer	Destination packets retransmitted or dropped
service	nominal	http, ftp, smtp, ssh, dns, ftp-data ,irc and (-) if not much used service
Sload	Float	Source bits per second
Dload	Float	Destination bits per second
Spkts	integer	Source to destination packet count
Dpkts	integer	Destination to source packet count
swin	integer	Source TCP window advertisement value
dwin	integer	Destination TCP window advertisement value
stcpb	integer	Source TCP base sequence number
dtcpb	integer	Destination TCP base sequence number
smeansz	integer	Mean of the how packet size transmitted by the src
dmeansz	integer	Mean of the how packet size transmitted by the dst
trans_depth	integer	Represents the pipelined depth into the connection of http request/response transaction
res_bdy_len	integer	Actual uncompressed content size of the data transferred from the server's http service.

Table 9 Number of system calls traces in different categories of AFDA-LD and AFDA-WD

ADFA- LD			ADFA-WD	
Dataset	Traces	System Calls	Traces	System Calls
Training data	833	308,077	355	13,504,419
Validation data	4372	2,122,085	1827	117,918,735
Attack data	746	317,388	5542	74,202,804
Total	5951	2,747,550	7724	205,625,958

CICIDS 2017

CICIDS2017 dataset comprises both benign behaviour and also details of new malware attacks: such as Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS (Sharafaldin et al., 2018). This dataset is labelled based on the timestamp, source and destination IPs, source and destination ports, protocols and attacks. A complete network topology was configured to collect this dataset which contains Modem, Firewall, Switches, Routers, and nodes with different operating systems (Microsoft Windows (like Windows 10, Windows 8, Windows 7, and Windows XP), Apple's macOS iOS, and open source operating system Linux). This dataset contains 80 network flow features from the captured network traffic.

Comparison of public IDS datasets

Since machine learning techniques are applied in AIDS, the datasets that are used for the machine learning techniques are very important to assess these techniques for realistic evaluation. Table 12 summarises popular public data sets, as well as some analysis techniques and results for each dataset from prior research. Table 13 summarizes the characteristics of the datasets.

Feature selection for IDS

Feature selection is helpful to decrease the computational difficulty, eliminate data redundancy, enhance the detection rate of the machine learning techniques, simplify data and reduce false alarms. In this line of research, some methods have been applied to develop a lightweight IDSs.

Feature selection techniques can be categorized into wrapper and filter methods. Wrapper methods estimate subgroups of variables to identify the feasible interactions between variables. There are two main drawbacks of these techniques: accumulative overfitting when the amount of data is insufficient and the important calculation time when the amount of variables is big.

Filter methods are normally applied as a pre-processing stage. The selection of features is separate of any machine learning techniques. As an alternative, features are nominated on the basis of their scores in several statistical tests for their correlation with the consequence variable.

As an example of the impact of feature selection on the performance of an IDS, consider the results in Table 14 which show the detection accuracy and time to build the IDS mode of the C4.5 classifier using the full dataset with 41 features of NSl-KDD dataset and with different features.

Table 11 ADFA-WD Vectors and Effects

Vectors
TCP ports - Web-based vectors; Browser attacks - Malware attachments.
Effects
Effects - Bind Shell - Reverse shell - Exploitation Remote operation - Staging - System manipulation Privilege escalation - Data exfiltration -Back-door insertion

Types of computer attacks

Cyber-attacks can be categorized based on the activities and targets of the attacker. Each attack type can be classified into one of the following four classes (Sung & Mukkamala, 2003):

- Denial-of-Service (DoS) attacks have the objective of blocking or restricting services delivered by the network, computer to the users.
- Probing attacks have the objective of acquisition of information about the network or the computer system.
- User-to-Root (U2R) attacks have the objective of a non-privileged user acquiring root or admin-user access on a specific computer or a system on which the intruder had user level access.
- Remote-to-Local (R2L) attacks involve sending packets to the victim machine. The cybercriminal

Table 10 ADFA-LD attack class

Attack	Payload	Vector	Count
Hydra-FTP	Password brute force	FTP by Hydra	162
Hydra-SSH	Password brute force	SSH Hydra	176
Adduser	Add new super user	Client-side poisoned executable	91
Java-Meterpreter	Java based Meterpreter	TlkiWiki vulnerability exploit	124
Meterpreter	Linux Meterpreter Payload	Client side poisoned executable	75
Webshell	C100 Webshell	PHP remote file inclusion vulnerability	118

Table 12 Comparison of results achieved by various methods on publically available IDS datasets

Dataset	Result	Observations	Reference
DARPA 98	Snort's detection, 69% of total generated alerts are considered to be false alarms.	SIDS is applied without AIDS	Hu, et al. (2009)
	ANN analysis system calls, 96% detection rate.	A classifier based on artificial neural network (ANN) has been executed for preparing and testing of framework.	McHugh (2000)
	SVM on subset of DARPA 98, 99.6% detection rate.	SVM isolates information into various classes by a hyperplane or hyperplanes since it can deal with multidimensional information. SVM usually demonstrate good performance for a binary class problem.	Chen, et al. (2005)
KDDCUP 99	Multivariate statistical analysis of audit data, 90% detection rate	Multivariate is used to reduce false alarm rates.	Ye, et al. (2002), Hotta, et al. (2008)
	The best results have been achieved by the C4.5 algorithm which attains the 95% true positive rate.	The decision trees created by C4.5 can be utilized for classification	Ferrari and Cribari-Neto (2004); Shafi and Abbass (2013); Laskov, et al. (2005)
	SMO classifier 97% detection rate.	This SVM based classifier with SMO implementation produces good detection accuracy. However, the accuracy reported is less than that in (Chen et al., 2005), because the KDDCUP 99 dataset is more complex and comprehensive than DARPA 98 dataset.	Shafi and Abbass (2013)
	The best model is an HNB model, where 95% confidence level is used to compare the models.	Hidden Naïve Bayes (HNB) techniques could be applied to IDS area that suffer from dimensionality, highly associated attributes and high network speed. HNB technique is better than the one based on the traditional NB method in terms of detection accuracy for IDS.	Koc, et al. (2012)
NSL-KDD	K-Nearest Neighbour (k-NN) algorithm, the detection rate of 94%.	The k-NN algorithm uses all labelled training instances as a model of the target function. During the classification phase, k-NN uses a similarity-based search strategy to determine a locally optimal hypothesis function.	Adebowale, et al. (2013)
	Naïve Bayes, the detection rate is 89%.	Bayesian classifiers provide moderate accuracy because the focus is on classifying the classes for the instances, not the exact probabilities.	Adebowale, et al. (2013)
	C4.5 gave the best detection rate of 99%.	C4.5 selects the feature of the data that most efficiently divides its set of samples into subsets, contributing to improved accuracy	Thaseen and Kumar (2013)
	SMO classifier, the detection rate is 97%.	The work also uses SVM based classifier and achieves detection rate similar to (Chen et al., 2005).	Adebowale, et al. (2013)
	Expectation Maximization (EM) clustering, the accuracy is 78%	EM forms a "soft" task of each row to various clusters in percentage to the probability of each cluster. The accuracy in this method is low as EM does not give a parameter covariance matrix for standard errors	Ahmed, et al. (2016)
ADFA-WD	Creech et al. have used Hidden Markov Model (HMM), Extreme Learning Machine (ELM) and SVM. They reported 74.3% accuracy for HMM, 98.57% accuracy for ELM and 99.64% accuracy for SVM.	The ADFA-WD is a much new data set and contains new attacks. This is why reported accuracy was not as good as for every machine learning technique when compared to the accuracy using legacy KDD98 data. SVM has been reported to produce the highest accuracy.	Creech and Hu (2014b)
ADFA-LD	100% accuracy for using ELM using original semantic feature	New semantic features are applied. Therefore, ELM, are capable to use the new semantic feature easily and quickly by including amounts of semantic phrases.	Creech and Hu (2014b)
CICIDS2017	94.5% accuracy obtained by using MLP solely, by using MLP and Payload Classifier together 95.2% accuracy rate is detected.	Feature selection is done by using Fisher Score algorithm.	Usteba, et al. (2018)
Bot-IoT	The highest accuracy from the SVM model. 98% detection rate	This SVM based method has produced good detection accuracy (Mitchell & Chen, 2015; Chen et al., 2005; Ferrari & Cribari-Neto, 2004)	Koroniotis, et al. (2018)

Table 13 Compassion of datasets (✓ = True, ✖ = False)

Dataset	Realistic Traffic	Label data	IoT traces	Zero-day attacks	Full packet captured	Year
DARPA 98	✓	✓	✖	✖	✓	1998
KDDCUP 99	✓	✓	✖	✖	✓	1999
CAIDA	✓	✖	✖	✖	✖	2007
NSL-KDD	✓	✓	✖	✖	✓	2009
ISCX 2012	✓	✓	✖	✖	✓	2012
ADFA-WD	✓	✓	✖	✓	✓	2014
ADFA-LD	✓	✓	✖	✓	✓	2014
CICIDS2017	✓	✓	✖	✓	✓	2017
Bot-IoT	✓	✓	✓	✓	✓	2018

learns the user's activities and obtains privileges which an end user could have on the computer system.

Within these broad categories, there are many different forms of computer attacks. A summary of these attacks with a brief explanation, characteristics, and examples are presented in Table 15.

IDS evasion techniques

This section discusses the techniques that a cybercriminal may use to avoid detection by IDS such as Fragmentation, Flooding, Obfuscation, and Encryption. These techniques pose a challenge for the current IDS as they circumvent existing detection methods.

Fragmentation

A packet is divided into smaller packets. The fragmented packets are then be reassembled by the recipient node at the IP layer before forwarding it to the Application layer. To examine fragmented traffic correctly, the network detector needs to assemble these fragments similarly as it was at fragmenting point. The restructuring of packets needs the detector to hold the data in memory and match the traffic against a signature database. Methods used by attackers to escape detection by hiding attacks as legitimate traffic are fragmentation overlap, overwrite, and timeouts (Ptacek & Newsham, 1998; Kolias et al., 2016). Fragmentation attack replaces information in the constituent fragmented packets with new information to generate a malicious packet.

Table 14 Detailed accuracy for C4.5 Decision tree classifier with different feature sets

Filter techniques	# of features	Accuracy	Time
Full set	41	99.55	2.76 Sec
Info Gain	13	99.64	0.84 Sec
Gain ratio	13	99.64	1.31 Sec
Chi-squared	13	99.65	0.92 Sec
Relief	13	99	0.93 Sec

Figure 8 shows the fragment overwrite. Packet Fragment 3 is generated by the attacker. The network intrusion detector must retain the state for all of the packets of the traffic which it is detecting.

The duration of time that the detector can maintain a state of traffic might be smaller than the period that the destination host can maintain a state of traffic (Xiong et al., 2017). The malware authors try to take advantage of any shortcoming in the detection method by delivering attack fragments over a long time.

Flooding

The attacker begins the attack to overwhelm the detector and this causes a failure of control mechanism. When the detector fails, all traffic would be allowed (Kolias et al., 2016). A popular method to create a flooding situation is spoofing the legitimate User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP). The traffic flooding is used to disguise the abnormal activities of the cybercriminal. Therefore, IDS would have extreme difficulty to find malicious packets in a huge amount of traffic.

Obfuscation

Obfuscation techniques can be used to evade detection, which are the techniques of concealing an attack by making the message difficult to understand (Kim et al., 2017). The terminology of obfuscation means changing the program code in a way that keeps it functionally identical with the aim to reduce detectability to any kind of static analysis or reverse engineering process and making it obscure and less readable. This obfuscation of malware enables it to evade current IDS.

Obfuscation attempts to utilize any limitations in the signature database and its capability to duplicate the way the computer host examines computer's data (Alazab & Khresiat, 2016). An effective IDS should be supporting the hexadecimal encoding format or having these hexadecimal strings in its set of attack signatures (Cova et al., 2010). Unicode/UTF-8 standard permits one character to be

Table 15 Classes of computer attacks

Types of Attack	Explanation	Example
Buffer Overflow	Attacks the buffer's boundaries and overwrites memory area.	Long URL strings are a common input. Cowan, et al. (1998)
Worm	Reproduces itself on the local host or through the network.	SQL Slammer, Mydoom, CodeRed Nimda.
Trojan	Programs appear attractive and genuine, but have malicious code embedded inside them.	Zeus, SpyEye Alazab, et al. (2013)
Denial of service (DoS)	A security event to disrupt the network services. It is started by forcing reset on the target computers. The users can no longer connect to the system because of unavailability of service.	Buffer overflow, Ping of death (PoD), TCP SYN, smurf, teardrop Zargar, et al. (2013)
Common Gateway Interface (CGI) Scripts	The attacker takes advantage of CGI scripts to create an attack by sending illegitimate inputs to the web server.	Phishing email; Aljawarneh (2016)
Traffic Flooding	Attacks the limited size of NIDS to handle huge traffic loads and to investigate for possible intrusions. If a cybercriminal can cause congestion in the networks, then NIDS will be busy in analyzing the traffic.	Denial of Service (Dos) or Distributed Denial of Service (DDoS) Zargar, et al. (2013)
Physical Attack	Aims to attack the physical mechanisms of the computer system.	Cold boot, evil maid (Pasqualetti et al., 2013).
Password Attack	Aims to break the password within a small time, and is noticed by a sequence of failures login.	A dictionary attack, Rainbow attack (Das et al., 2014).
Information Gathering	Gathers information or finds weaknesses in computers or networks by sniffing or searching.	System scan, port scan, (Bou-Harb et al., 2014).
User to Root (U2R) attack	The cybercriminal accesses as a normal user in the beginning and then upgrades to a super-user which may lead to exploitation of several vulnerabilities of the system.	Intercept packets, rainbow attack, social engineering Rootkit, load module, (Perl Raiyn, 2014).
Remote to Local (R2L) attack	The cybercriminal sends packets to a remote system by connecting to the network without having an account on the system.	Warezclient, ftp write, multihop,phf, spy, warezmaster, imap (Raiyn, 2014).
Probe	Identifying the valid IP addresses by scanning the network to gather host data packets.	Sweep, portsweep (So-In et al., 2014)

symbolized in several various formats. Cybercriminals may also use double-encoded data, exponentially escalating the number of signatures required to detect the attack.

SIDS relies on signature matching to identify malware where the signatures are created by human experts by translating a malware from machine code into a symbolic language such as Unicode. However, the use of code obfuscation is very valuable for cybercriminals to avoid IDSs.

Encryption

Generally, encryption offers a number of security services, such as data confidentiality, integrity, and privacy. Malware authors employ these security attributes to escape detection and conceal attacks that may target a computer system. For example, attacks on encrypted protocols such as HyperText Transfer Protocol Secure (HTTPS) cannot be read by an IDS (Metke & Ekl, 2010). The IDS cannot match the encrypted traffic to the existing Database signatures if it doesn't interpret the encrypted traffic. Therefore, examining encrypted traffic makes it difficult for detectors to detect attacks (Butun et al., 2014). For example, packet content-based features have been applied extensively to identify malware from normal traffic, which cannot readily be applied if the packet is encrypted.

These challenges motivate investigators to use some statistical network flow features, which do not rely on packet content (Camacho et al., 2016). As a result of this, malware can potentially be identified from normal traffic.

Challenges of IDS

Although there has been a lot of research on IDSs, many essential matters remain. IDSs have to be more accurate, with the capability to detect a varied ranging of intrusions with fewer false alarms and other challenges.

Challenges of IDS for ICSs

Industrial Control Systems (ICSs) are commonly comprised of two components: Supervisory Control and Data Acquisition (SCADA) hardware which receives information from sensors and then controls the mechanical machines; and the software that enables human administrators to control the machines.

Cyber attacks on ICSs is a great challenge for the IDS due to unique architectures of ICSs as the attackers are currently focusing on ICSs. A standout amongst the recent attacks against ICSs is the Stuxnet attack, which is known as the first cyber-warfare weapon. Dissimilar to a typical attack, the primary target of Stuxnet was probably

Fig. 8 Fragment – Overwrite

the Iranian atomic program (Nourian & Madnick, 2018). Attacks that could target ICSs could be state-sponsored or they might be launched by the competitors, internals attackers with a malicious target, or even hacktivists.

The potential consequences of compromised ICS can be devastating to public health and safety, national security, and the economy. Compromised ICS systems have led to the extensive cascading power outages, dangerous toxic chemical releases, and explosions. It is therefore important to use secure ICSs for reliable, safe, and flexible performance.

It is critical to have IDS for ICSs that takes into account unique architecture, realtime operation and dynamic environment to protect the facilities from the attacks. Some critical attacks on ICSs are given below:

- In 2008, Conficker malware infected ICS systems, such as an aeroplane's internal systems. Conficker disables many security features and automatic backup settings, erases stored data and opens associations to get commands from a remote PC (Pretorius & van Niekerk, 2016).
- In 2009, a 14-year-old schoolboy hacked the city's tram system and used a homemade remote device to redirect a number of trams, injuring 12 passengers (Rege-Patwardhan, 2009).
- In 2017, WannaCry ransomware spread globally and seriously effected the National Health System, UK and prevented emergency clinic specialists from using health systems (Mohurle & Patil, 2017).

Since Microsoft no longer creates security patches for legacy systems, they can simply be attacked by new types of ransomware and zero-day malware.

Similiarly, it may not be possible to fix or update the operating systems of ICSs for legacy applications.

A robust IDS can help industries and protect them from the threat of cyber attacks. Unfortunately, current intrusion detection techniques proposed in the literature focus at the software level. A vital detection approach is needed to detect the zero-day and complex attacks at the software level as well as at hardware level without any previous knowledge. This can be done by integrating both hardware and software intrusion detection systems and extracting useful features of both HIDS and NIDS.

Challenge of IDS on intrusion evasion detection

Detecting attacks masked by evasion techniques is a challenge for both SIDS and AIDS. The ability of evasion techniques would be determined by the ability of IDS to bring back the original signature of the attacks or create new signatures to cover the modification of the attacks. Robustness of IDS to various evasion techniques still needs further investigation. For example, SIDS in regular expressions can detect the deviations from simple mutation such as manipulating space characters, but they are still useless against a number of encryption techniques.

Discussion and conclusion

Cybercriminals are targeting computer users by using sophisticated techniques as well as social engineering strategies. Some cybercriminals are becoming increasingly sophisticated and motivated. Cybercriminals have shown their capability to obscure their identities, hide their communication, distance their identities from illegal profits, and use infrastructure that is resistant to compromise. Therefore, it becomes increasingly important for computer systems to be protected using advanced intrusion detection systems which are capable of detecting modern malware. In order to design and build such IDS systems, it is necessary to have a complete overview of the strengths and limitations of contemporary IDS research.

In this paper, we have presented, in detail, a survey of intrusion detection system methodologies, types, and technologies with their advantages and limitations. Several machine learning techniques that have been proposed to detect zero-day attacks are reviewed. However, such approaches may have the problem of generating and updating the information about new attacks and yield high false alarms or poor accuracy. We summarized the results of recent research and explored the contemporary models on the performance improvement of AIDS as a solution to overcome on IDS issues.

In addition, the most popular public datasets used for IDS research have been explored and their data collection techniques, evaluation results and limitations have been discussed. As normal activities are frequently changing and may not remain effective over time, there exists a need for newer and more comprehensive datasets that contain wide-spectrum of malware activities. A new malware dataset is needed, as most of the existing machine

learning techniques are trained and evaluated on the knowledge provided by the old dataset such as DARPA/KDD99, which do not include newer malware activities. Therefore, testing is done using these dataset collected in 1999 only, because they are publicly available and no other alternative and acceptable datasets are available. While widely accepted as benchmarks, these datasets no longer represent contemporary zero-day attacks. Though ADFA dataset contains many new attacks, it is not adequate. For that reason, testing of AIDS using these datasets does not offer a real evaluation and could result in inaccurate claims for their effectiveness.

This study also examines four common evasion techniques to determine their ability to evade the recent IDSs. An effective IDS should be able to detect different kinds of attacks accurately including intrusions that incorporate evasion techniques. Developing IDSs capable of overcoming the evasion techniques remains a major challenge for this area of research.

Acknowledgments
The research is supported by the Internet Commerce Security Laboratory, Federation University Australia. The authors are grateful to the Centre for Informatics and Applied Optimization (CIAO) for their support.

Authors' contributions
AK has participated presented, in detail, a survey of intrusion detection system methodologies, types, and technologies with their advantages and limitations. Several machine learning techniques have been proposed to detect zero-day attacks are reviewed. IG, PV, and JK have gone through the article. All authors read and approved the final manuscript.

Funding
This work was carried out within the Internet Commerce Security Lab, which is funded by Westpac Banking Corporation.

Availability of data and materials
This manuscript has not been published and is not under consideration for publication elsewhere.

Competing interests
The authors declare that they have no competing interests.

Received: 29 October 2018 Accepted: 25 June 2019
Published online: 17 July 2019

References
A. Abbasi, J. Wetzels, W. Bokslag, E. Zambon, and S. Etalle, "On emulation-based network intrusion detection systems," in Research in attacks, intrusions and defenses: 17th international symposium, RAID 2014, Gothenburg, Sweden, September 17–19, 2014. Proceedings, A. Stavrou, H. Bos, and G. Portokalidis, Eds. Cham: Springer International Publishing, 2014, pp. 384–404

A. A. Aburomman and M. B. Ibne Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," Appl Soft Comput, vol. 38, pp. 360–372, 2016/01/01/ 2016

Adebowale A, Idowu S, Amarachi AA (2013) Comparative study of selected data mining algorithms used for intrusion detection. International Journal of Soft Computing and Engineering (IJSCE) 3(3):237–241

Agrawal S, Agrawal J (2015) Survey on anomaly detection using data mining techniques. Procedia Computer Science 60:708–713

M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," J Netw Comput Appl, vol. 60, pp. 19–31, 1// 2016

A. Alazab, J. Abawajy, M. Hobbs, R. Layton, and A. Khraisat, "Crime toolkits: the Productisation of cybercrime," in 2013 12th IEEE international conference on trust, security and privacy in computing and communications, 2013, pp. 1626–1632

A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, "Using feature selection for intrusion detection system," in 2012 international symposium on communications and information technologies (ISCIT), 2012, pp. 296–301

Alazab A, Hobbs M, Abawajy J, Khraisat A, Alazab M (2014) Using response action with intelligent intrusion detection and prevention system against web application malware. Information Management & Computer Security 22(5):431–449

Alazab A, Khresiat A (2016) New strategy for mitigating of SQL injection attack. Int J Comput Appl 154(11)

Alcaraz C (2018) Cloud-assisted dynamic resilience for cyber-physical control systems. IEEE Wirel Commun 25(1):76–82

S. A. Aljawarneh, "Emerging challenges, security issues, and Technologies in Online Banking Systems," Online Banking Security Measures and Data Protection, p. 90, 2016

C. Annachhatre, T. H. Austin, and M. Stamp, "Hidden Markov models for malware classification," Journal of Computer Virology and Hacking Techniques, vol. 11, no. 2, pp. 59–73, 2015/05/01 2015

Ara A, Louzada F, Diniz CAR (2017) Statistical monitoring of a web server for error rates: a bivariate time-series copula-based modeling approach. J Appl Stat:1–14

Ashfaq RAR, Wang X-Z, Huang JZ, Abbas H, He Y-L (2017) Fuzziness based semi-supervised learning approach for intrusion detection system. Inf Sci 378:484–497

Australian. (2017, November). Australian cyber security center threat report 2017. Available: https://www.acsc.gov.au/publications/ACSC_Threat_Report_2017.pdf

S. Axelsson, "Intrusion detection systems: a survey and taxonomy," technical report 2000

Bajaj K, Arora A (2013) Dimension reduction in intrusion detection features using discriminative machine learning approach. IJCSI International Journal of Computer Science Issues 10(4):324–328

Bhuyan MH, Bhattacharyya DK, Kalita JK (2014) Network anomaly detection: methods, systems and tools. IEEE Communications Surveys & Tutorials 16(1):303–336

J. J. Blount, D. R. Tauritz, and S. A. Mulder, "Adaptive rule-based malware detection employing learning classifier systems: a proof of concept," in Computer software and applications conference workshops (COMPSACW), 2011 IEEE 35th annual, 2011, pp. 110–115: IEEE

Bou-Harb E, Debbabi M, Assi C (2014) Cyber scanning: a comprehensive survey. IEEE Communications Surveys & Tutorials 16(3):1496–1519

Breach_LeveL_Index. (2017, November). Data breach statistics. Available: http://breachlevelindex.com/

Breiman L (1996) Bagging predictors. Machine Learning, journal article 24(2):123–140

Buczak AL, Guven E (2016) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials 18(2):1153–1176

Butun I, Morgera SD, Sankar R (2014) A survey of intrusion detection systems in wireless sensor networks. IEEE Communications Surveys & Tutorials 16(1):266–282

J. Camacho, A. Pérez-Villegas, P. García-Teodoro, and G. Maciá-Fernández, "PCA-based multivariate statistical network monitoring for anomaly detection," Computers & Security, vol. 59, pp. 118–137, 6// 2016

O. Can and O. K. Sahingoz, "A survey of intrusion detection systems in wireless sensor networks," in 2015 6th international conference on modeling, simulation, and applied optimization (ICMSAO), 2015, pp. 1–6

L. Chao, S. Wen, and C. Fong, "CANN: an intrusion detection system based on combining cluster centers and nearest neighbors," Knowl-Based Syst, vol. 78, pp. 13–21, 4// 2015

S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," Computers & Security, vol. 24, no. 4, pp. 295–307, 6// 2005

W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for intrusion detection," Comput Oper Res, vol. 32, no. 10, pp. 2617–2634, 2005/10/01/ 2005

M. Cova, C. Kruegel, and G. Vigna, "Detection and analysis of drive-by-download attacks and malicious JavaScript code," Presented at the Proceedings of the 19th international conference on world wide web, Raleigh, North Carolina, USA, 2010

C. Cowan *et al.*, "Stackguard: automatic adaptive detection and prevention of buffer-overflow attacks," in USENIX security symposium, 1998, vol. 98, pp. 63–78: San Antonio, TX

G. Creech, "Developing a high-accuracy cross platform host-based intrusion detection system capable of reliably detecting zero-day attacks," University of New South Wales, Canberra, Australia, 2014

Creech G, Hu J (2014a) A semantic approach to host-based intrusion detection systems using Contiguousand Discontiguous system call patterns. IEEE Trans Comput 63(4):807–819
Creech G, Hu J (2014b) A semantic approach to host-based intrusion detection systems using contiguous and Discontiguous system call patterns. IEEE Trans Comput 63(4):807–819
A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The tangled web of password reuse," in NDSS, 2014, vol. 14, pp. 23–26
H. Debar, M. Dacier, and A. Wespi, "A revised taxonomy for intrusion-detection systems," in *Annales des télécommunications*, 2000, vol. 55, no. 7–8, pp. 361–378: Springer
Z. Du, K. Palem, A. Lingamneni, O. Temam, Y. Chen, and C. Wu, "Leveraging the error resilience of machine-learning applications for designing highly energy efficient accelerators," in 2014 19th Asia and South Pacific design automation conference (ASP-DAC), 2014, pp. 201–206
S. Dua and X. Du, Data mining and machine learning in cybersecurity. CRC press, 2016
S. Duque and M. N. b. Omar, "Using data mining algorithms for developing a model for intrusion detection system (IDS)," Procedia Computer Science, vol. 61, no. Supplement C, pp. 46–51, 2015/01/01/ 2015
S. Elhag, A. Fernández, A. Bawakid, S. Alshomrani, and F. Herrera, "On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems," Expert Syst Appl, vol. 42, no. 1, pp. 193–202, 1// 2015
D. M. Farid, N. Harbi, and M. Z. Rahman, "Combining naive bayes and decision tree for adaptive intrusion detection," arXiv preprint arXiv:1005.4496, 2010
S. L. P. Ferrari and F. Cribari-Neto, J Appl Stat, vol. 31, no. null, p. 799, 2004
M. Goldstein, "FastLOF: an expectation-maximization based local outlier detection algorithm," in Pattern recognition (ICPR), 2012 21st international conference on, 2012, pp. 2282–2285: IEEE
Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD explorations newsletter 11(1):10–18
Hendry G, Yang S (2008) Intrusion signature creation via clustering anomalies
P. Hick, E. Aben, K. Claffy, and J. Polterock, "the CAIDA DDoS attack 2007 dataset," ed, 2007
Hoque MAM, Bikas MAN (2012) An implementation of intrusion detection system using genetic algorithm. International Journal of Network Security & Its Applications 4:2
L. K. Hotta, E. C. Lucas, and H. P. Palaro, Multinat. Financ J, vol 12, no null, p. 205, 2008
Hu J, Yu X, Qiu D, Chen HH (2009) A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection. IEEE Netw 23(1):42–47
Hu W, Gao J, Wang Y, Wu O, Maybank S (2014) Online Adaboost-based parameterized methods for dynamic distributed network intrusion detection. IEEE Transactions on Cybernetics 44(1):66–82
N. Hubballi and V. Suryanarayanan, "False alarm minimization techniques in signature-based intrusion detection systems: a survey," Comput Commun, vol. 49, pp. 1–17, 8/1/ 2014
M. A. Jabbar, R. Aluvalu, and S. S. Reddy S, "RFAODE: A Novel Ensemble Intrusion Detection System," *Procedia Computer Science*, vol. 115, pp. 226–234, 2017/01/01/ 2017
S.-Y. Ji, B.-K. Jeong, S. Choi, and D. H. Jeong, "A multi-level intrusion detection method for abnormal network behaviors," J Netw Comput Appl, vol. 62, no. Supplement C, pp. 9–17, 2016/02/01/ 2016
KDD. (1999, June). *The 1999 KDD intrusion detection*. Available: http://kdd.ics.uci.edu/databases/kddcup99/task.html
Kenkre PS, Pai A, Colaco L (2015a) Real time intrusion detection and prevention system. In: Satapathy SC, Biswal BN, Udgata SK, Mandal JK (eds) Proceedings of the 3rd international conference on Frontiers of intelligent computing: theory and applications (FICTA) 2014: volume 1. Springer International Publishing, Cham, pp 405–411
Kenkre PS, Pai A, Colaco L (2015b) Real Time Intrusion Detection and Prevention System. Springer International Publishing, Cham, pp 405–411
Khraisat A, Gondal I, Vamplew P (2018) An anomaly intrusion detection system using C5 decision tree classifier. In: Trends and applications in knowledge discovery and data mining. Springer International Publishing, Cham, pp 149–155
D. Kim *et al.*, "DynODet: detecting dynamic obfuscation in malware," in Detection of intrusions and malware, and vulnerability assessment: 14th international conference, DIMVA 2017, Bonn, Germany, July 6–7, 2017, Proceedings, M. Polychronakis and M. Meier, Eds. Cham: Springer International Publishing, 2017, pp. 97–118
G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," Expert Syst Appl, vol. 41, no. 4, Part 2, pp. 1690–1700, 2014/03/01/ 2014
L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a hidden Naïve Bayes multiclass classifier," Expert Syst Appl, vol. 39, no. 18, pp. 13492–13500, 2012/12/15/ 2012
Kolias C, Kambourakis G, Stavrou A, Gritzalis S (2016) Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Communications Surveys & Tutorials 18(1):184–208
N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: bot-IoT dataset," arXiv preprint arXiv:*1811.00701*, 2018
Kreibich C, Crowcroft J (2004) Honeycomb: creating intrusion detection signatures using honeypots. SIGCOMM Comput Commun Rev 34(1):51–56
Kshetri N, Voas J (2017) Hacking power grids: a current problem. Computer 50(12):91–95
P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning intrusion detection: supervised or unsupervised?," in Image analysis and processing – ICIAP 2005: 13th international conference, Cagliari, Italy, September 6–8, 2005. Proceedings, F. Roli and S. Vitulano, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 50–57
Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," Expert Syst Appl, vol. 39, no. 1, pp. 424–430, 2012/01/01/ 2012
Liao H-J, Lin C-HR, Lin Y-C, Tung K-Y (2013b) Intrusion detection system: a comprehensive review. J Netw Comput Appl 36(1):16–24
H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: a comprehensive review," J Netw Comput Appl, vol. 36, no. 1, pp. 16–24, 2013a/01/01/ 2013
Lin C, Lin Y-D, Lai Y-C (2011) A hybrid algorithm of backward hashing and automaton tracking for virus scanning. IEEE Trans Comput 60(4):594–601
W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: an intrusion detection system based on combining cluster centers and nearest neighbors," Knowl-Based Syst, vol. 78, no. Supplement C, pp. 13–21, 2015/04/01/ 2015
Liu X, Zhu P, Zhang Y, Chen K (2015) A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure. IEEE Transactions on Smart Grid 6(5):2435–2443
T. F. Lunt, "Automated audit trail analysis and intrusion detection: a survey," in Proceedings of the 11th National Computer Security Conference, 1988, vol. 353: Baltimore, MD
J. Lyngdoh, M. I. Hussain, S. Majaw, and H. K. Kalita, "An intrusion detection method using artificial immune system approach," in International conference on advanced informatics for computing research, 2018, pp. 379–387: Springer
McHugh J (2000) Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Trans Inf Syst Secur 3(4):262–294
C. R. Meiners, J. Patel, E. Norige, E. Torng, and A. X. Liu, "Fast regular expression matching using small TCAMs for network intrusion detection and prevention systems," presented at the Proceedings of the 19th USENIX conference on security, Washington, DC, 2010
Meshram A, Haas C (2017) Anomaly detection in industrial networks using machine learning: a roadmap. In: Beyerer J, Niggemann O, Kühnert C (eds) Machine learning for cyber physical systems: selected papers from the international conference ML4CPS 2016. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 65–72
Metke AR, Ekl RL (2010) Security Technology for Smart Grid Networks. IEEE Transactions on Smart Grid 1(1):99–107
MIT Lincoln Laboratory. (1999, June). DARPA Intrusion Detection Data Sets. Available: https://www.ll.mit.edu/ideval/data/
Mitchell R, Chen IR (2015) Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems. IEEE Transactions on Dependable and Secure Computing 12(1):16–30
C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," J Netw Comput Appl, vol. 36, no. 1, pp. 42–57, 2013/01/01/ 2013
Mohurle S, Patil M (2017) A brief study of wannacry threat: ransomware attack 2017. Int J Adv Res Comput Sci 8(5)
S. N. Murray, B. P. Walsh, D. Kelliher, and D. T. J. O'Sullivan, "Multi-variable optimization of thermal energy efficiency retrofitting of buildings using static modelling and genetic algorithms – a case study," Build Environ, vol. 75, no. Supplement C, pp. 98–107, 2014/05/01/ 2014

Nourian A, Madnick S (2018) A systems theoretic approach to the security threats in cyber physical systems applied to Stuxnet. IEEE Transactions on Dependable and Secure Computing 15(1):2–13
Pasqualetti F, Dörfler F, Bullo F (2013) Attack detection and identification in cyber-physical systems. IEEE Trans Autom Control 58(11):2715–2729
A. Patel, M. Taghavi, K. Bakhtiyari, and J. Celestino Júnior, "An intrusion detection and prevention system in cloud computing: a systematic review," J Netw Comput Appl, vol. 36, no. 1, pp. 25–41, 2013/01/01/ 2013
Pretorius B, van Niekerk B (2016) Cyber-security for ICS/SCADA: a south African perspective. International Journal of Cyber Warfare and Terrorism (IJCWT) 6(3):1–16
T. H. Ptacek and T. N. Newsham, "Insertion, evasion, and denial of service: eluding network intrusion detection," DTIC Document 1998
W. Qingtao and S. Zhiqing, "Network anomaly detection using time series analysis," in Joint international conference on autonomic and autonomous systems and international conference on networking and services - (icas-isns'05), 2005, pp. 42–42
Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106
J. R. Quinlan, C4. 5: programs for machine learning. Elsevier, 2014
Raiyn J (2014) A survey of cyber attack detection strategies. International Journal of Security and Its Applications 8(1):247–256
Rath PS, Barpanda NK, Singh R, Panda S (2017) A prototype Multiview approach for reduction of false alarm rate in network intrusion detection system. Int J Comput Netw Commun Secur 5(3):49
Rege-Patwardhan A (2009) Cybercrimes against critical infrastructures: a study of online criminal organization and techniques. Crim Justice Stud 22(3):261–271
K. Riesen and H. Bunke, "IAM graph database repository for graph based pattern recognition and machine learning," in Structural, syntactic, and statistical pattern recognition: joint IAPR international workshop, SSPR & SPR 2008, Orlando, USA, December 4–6, 2008. Proceedings, N. da Vitoria Lobo et al., Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 287–297
Roesch M (1999) Snort-lightweight intrusion detection for networks. In: Proceedings of the 13th USENIX conference on system administration. Seattle, Washington, pp 229–238
Rutkowski L, Jaworski M, Pietruczuk L, Duda P (2014) Decision trees for mining data streams based on the Gaussian approximation. IEEE Trans Knowl Data Eng 26(1):108–119
Sadotra P, Sharma C (2016) A survey: intelligent intrusion detection system in computer security. Int J Comput Appl 151(3):18–22
Sadreazami H, Mohammadi A, Asif A, Plataniotis KN (2018) Distributed-graph-based statistical approach for intrusion detection in cyber-physical systems. IEEE Transactions on Signal and Information Processing over Networks 4(1):137–147
Shafi K, Abbass HA (2013) Evaluation of an adaptive genetic-based signature extraction system for network intrusion detection. Pattern Analysis and Applications, journal article 16(4):549–566
Shakshuki EM, Kang N, Sheltami TR (2013) A secure intrusion-detection system for MANETs. IEEE Trans Ind Electron 60(3):1089–1098
I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in ICISSP, 2018, pp. 108–116
Shen C, Liu C, Tan H, Wang Z, Xu D, Su X (2018) Hybrid-augmented device fingerprinting for intrusion detection in industrial control system networks. IEEE Wirel Commun 25(6):26–31
Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA (2012) Toward developing a systematic approach to generate benchmark datasets for intrusion detection. computers & security 31(3):357–374
C. So-In, N. Mongkonchai, P. Aimtongkham, K. Wijitsopon, and K. Rujirakul, "An evaluation of data mining classification models for network intrusion detection," in 2014 fourth international conference on digital information and communication technology and its applications (DICTAP), 2014, pp. 90–94
P. Stavroulakis and M. Stamp, Handbook of information and communication security. Springer Science & Business Media, 2010
Studnia I, Alata E, Nicomette V, Kaâniche M, Laarouchi Y (2018) A language-based intrusion detection approach for automotive embedded networks. Int J Embed Syst 10(1):1–12
Subramanian S, Srinivasan VB, Ramasa C (2012) Study on classification algorithms for network intrusion systems. Journal of Communication and Computer 9(11):1242–1246
A. H. Sung and S. Mukkamala, "Identifying important features for intrusion detection using support vector machines and neural networks," in Symposium on Applications and the Internet, 2003, pp. 209–216
Symantec, "Internet security threat report 2017," April, 7017 2017, vol. 22 Available: https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf
Tan Z, Jamdagni A, He X, Nanda P, Liu RP (2014) A system for denial-of-service attack detection based on multivariate correlation analysis. IEEE Transactions on Parallel and Distributed Systems 25(2):447–456
M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in 2009 IEEE symposium on computational intelligence for security and defense applications, 2009, pp. 1–6
S. Thaseen and C. A. Kumar, "An analysis of supervised tree based classifiers for intrusion detection system," in 2013 international conference on pattern recognition, informatics and Mobile engineering, 2013, pp. 294–299
S. Ustebay, Z. Turgut, and M. A. Aydin, "Intrusion detection system with recursive feature elimination by using random Forest and deep learning classifier," in 2018 international congress on big data, deep learning and fighting cyber terrorism (IBIGDELFT), 2018, pp. 71–76
Vigna G, Kemmerer RA (1999) NetSTAT: a network-based intrusion detection system. J Comput Secur 7:37–72
J. Viinikka, H. Debar, L. Mé, A. Lehikoinen, and M. Tarvainen, "Processing intrusion detection alert aggregates with time series modeling," Information Fusion, vol. 10, no. 4, pp. 312–324, 2009/10/01/ 2009
D. Wagner and P. Soto, "Mimicry attacks on host-based intrusion detection systems," presented at the Proceedings of the 9th ACM conference on computer and communications security, Washington, DC, USA, 2002
N. Walkinshaw, R. Taylor, and J. Derrick, "Inferring extended finite state machine models from software executions," *Empirical Software Engineering,* journal article vol. 21, no. 3, pp. 811–853, June 01 2016
G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," Expert Syst Appl, vol. 37, no. 9, pp. 6225–6232, 2010/09/01/ 2010
L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning," arXiv preprint arXiv:1801.06275, 2018
Xiong Q, Xu Y, Zhang B f, Wang F (2017) Overview of the evasion resilience testing Technology for Network Based Intrusion Protecting Devices. In: 2017 IEEE 18th international symposium on high assurance systems engineering (HASE), pp 146–152
X. Yang and Y. L. Tian, "EigenJoints-based action recognition using Naïve-Bayes-nearest-neighbor," in 2012 IEEE computer society conference on computer vision and pattern recognition workshops, 2012, pp. 14–19
Ye N, Emran SM, Chen Q, Vilbert S (2002) Multivariate statistical analysis of audit trails for host-based intrusion detection. IEEE Trans Comput 51(7):810–820
Y. Yuan, G. Kaklamanos, and D. Hogrefe, "A novel semi-supervised Adaboost technique for network anomaly detection," Presented at the Proceedings of the 19th ACM international conference on modeling, analysis and simulation of wireless and Mobile systems, Malta, Malta, 2016
Zargar J, Tipper (2013) A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Communications Surveys & Tutorials 15(4):2046–2069

Publisher's Note