Telecommunication Systems (2019) 70:447–489
https://doi.org/10.1007/s11235-018-0475-8

A comprehensive survey on network anomaly detection

Gilberto Fernandes Jr.[1] · Joel J. P. C. Rodrigues[1,2,3,4,6] · Luiz Fernando Carvalho[5] · Jalal F. Al-Muhtadi[6] · Mario Lemes Proença Jr.[7]

Published online: 2 July 2018

Abstract
Nowadays, there is a huge and growing concern about security in information and communication technology among the scientific community because any attack or anomaly in the network can greatly affect many domains such as national security, private data storage, social welfare, economic issues, and so on. Therefore, the anomaly detection domain is a broad research area, and many different techniques and approaches for this purpose have emerged through the years. In this study, the main objective is to review the most important aspects pertaining to anomaly detection, covering an overview of a background analysis as well as a core study on the most relevant techniques, methods, and systems within the area. Therefore, in order to ease the understanding of this survey's structure, the anomaly detection domain was reviewed under five dimensions: (1) network traffic anomalies, (2) network data types, (3) intrusion detection systems categories, (4) detection methods and systems, and (5) open issues. The paper concludes with an open issues summary discussing presently unsolved problems, and final remarks.

Keywords Anomaly detection · Network security · Network management · Intrusion detection · Anomaly detection methods

1 Introduction

Nowadays, the scientific community has a constant worry about high-efficiency security and quality of service in large-scale networks. The expansion of new communication technologies and services, along with an increasing number of interconnected network devices, web users, services, and applications, contributes to making computer networks ever larger and more complex as systems. Moreover, there is the so called *boundless communication paradigm*, for next generation networks, which envisages offering *anytime*, *anywhere*, *anyhow* communications to its users and requires the full integration and interoperability of emergent technologies [1–4]. These issues make it even more complex and challenging to maintain precise network management and lead to serious network vulnerabilities, as security incidents may occur more frequently [5,6].

Such security instances can be caused either by outsiders, as malicious attacks aiming to shut down services or steal private information, or by inside factors (operational problems), such as configuration errors, server crashes, power outages, traffic congestion, or non-malicious large file transfers [7]. Regardless of the source, such threats, which are commonly called anomalies, can have a significant impact

✉ Joel J. P. C. Rodrigues
joeljr@ieee.org

Gilberto Fernandes Jr.
gilfernandes6@gmail.com

Luiz Fernando Carvalho
luizfcarvalhoo@gmail.com

Jalal F. Al-Muhtadi
jalal@ccis.edu.sa

Mario Lemes Proença Jr.
proenca@uel.br

1 Instituto de Telecomunicações, Universidade da Beira Interior, Covilhã, Portugal

2 National Institute of Telecommunications (Inatel), Av. João de Camargo, 510 – Center, Santa Rita do Sapucaì 37540-000, Brazil

3 ITMO University, St. Petersburg, Russia

4 University of Fortaleza (UNIFOR), Fortaleza, Brazil

5 State University of Campinas (UNICAMP), Campinas, Brazil

6 College of Computer and Information Sciences (CCIS), King Saud University (KSU), Riyadh 12372, Saudi Arabia

7 Computer Science Department, State University of Londrina (UEL), Londrina, Brazil

on the network service and end-users and harm computer network operations and availability.

The term anomaly has several definitions. Barnett and Lewis define a data set anomaly as "observation (or a subset of observations) which appears to be inconsistent with the remainder of that set of data" [8]. Chandola et al. express this term as "patterns in data not conforming to a well-defined notion of normal behavior" [9]. According to Lakhina et al., "anomalies are unusual and significant changes in a network's traffic levels, which can often span multiple links" [10]. Hoque et al. define it as "non-conforming interesting patterns compared to the well-defined notion of normal behavior" [11]. By these definitions, it is clear that the concept of normality is one of the main steps toward developing a solution to detect network anomalies.

Although apparently unpretentious, the problem of defining a region denoting normal behavior and marking as an anomaly any occasion contrasting this normal pattern, is defiant. Faster diagnosis, lower complexity and suitable corrections of the causes are the main objectives of the field. Every factor is vital to developing a better anomaly detection approach. The precision and speed factors, alongside with the correct identification of such abnormal events in a timely fashion are critical to reducing significant service degradation, malicious damage, and cost. For this reason, the research community has been developing a lot of models, algorithms, and mechanisms, over the years, to develop better solutions and approaches to guaranteeing the health of ever larger and complex network systems.

Researchers have been studying the anomaly detection subject since the early 19th century, and so far, they have produced a multitude of papers, each using a variety of techniques, from statistical models, up to evolutionary computation approaches. Nevertheless, it is not a straightforward task to identify and categorize all existing anomaly detection techniques. Plenty of topics must be considered, such as anomaly types, system types, techniques and algorithms used, as well as technical dilemmas such as processing costs and network complexity. Therefore, this leads to the fragmented literature available today, in which many works try to summarize everything but are unable to show the bigger picture of the anomaly detection spectrum.

As in [12,13], the focus is just on the most popular techniques and methods, such as machine learning, clustering and statistical approaches. Still, surveys such as [14,15] briefly discuss the whole problem statement, setting aside relevant topics such as data set, challenges, and recommendations. Marnerides et al. [16] have a reviewed anomaly detection over backbone networks. Although each of those inspected surveys summarizes many important topics pertaining to anomaly detection, they are not entirely complete. For instance, some of them emphasize anomaly types but do not cover all kinds of methods while others research upon vast approaches but forget about the basis of intrusion detection systems and data input, and so on. For this reason, in this survey, we present a systematic overview of the whole anomaly detection domain under five dimensions: (i) network anomalies, (ii) network data types, (iii) intrusion detection systems overview, (iv) detection methods and systems, and (v) open issues. Table 1 provides a comparison between some anomaly detection surveys with regard to the variety of techniques they address.

At last, this survey aims to bring a complete and straightforward review of state-of-the-art anomaly detection topic. Then, the main contributions of the paper are the following:

- Review the anomaly detection subject under five research directions;
- A detailed study of the most relevant techniques, methods, and systems within the area;
- Address the main drawbacks found in the analyzed surveys extracted from the literature;
- Analysis of the four traffic anomaly types categorized by the causal aspect;
- Forward-looking discussion and comparative analysis of other surveys regarding open issues and future trends.

This paper is organized as follows. The introduction presents the overall motivation for developing this survey and a comparison with other surveys in the literature. Section 2 defines, categorizes, explains, and provides examples of most common types of network anomalies. Section 3 gives a brief explanation of network data types used as input in anomaly detection systems. Section 4 gives a complete overview of intrusion detection systems and the differences between each approach. Section 5 is the core section, which lists many anomaly detection methods and systems using a variety of techniques and algorithms of different nature and purpose. Section 6 summarizes everything discussed in previous sections into some topics considered as open challenges in the anomaly detection domain. Finally, Sect. 7 concludes the survey. Figure 1 shows all contents presented and discussed within the paper.

2 Network traffic anomalies

One of the first tasks envisioned by researchers in creating an anomaly detection model is the correct identification and definition of the problem statement. It means that there must be prior knowledge about what type of anomaly researchers would deal with. There are several types of network traffic anomalies, and each author surveying this topic addresses them differently. For the sake of simplicity, and after analyz-

Table 1 A comparison between anomaly detection surveys

Content		Surveys								
		Patcha and Park [13]	Chandola et al. [9]	Weiyu et al. [15]	Thottan et al. [12]	Yu et al. [14]	Bhuyan et al. [17]	Marnerides et al. [16]	Ahmed et al. [18]	This survey
Year		2007	2009	2009	2010	2012	2014	2014	2016	2018
Traffic anomalies by nature	Point		✓				✓		✓	✓
	Collective		✓				✓		✓	✓
	Contextual		✓				✓		✓	✓
Traffic anomalies by causal aspect	Operational									✓
	Flash Crowd									✓
	Measurement									✓
	Network attack						✓	✓	✓	✓
Network data types	TCP dump									✓
	SNMP				✓					✓
	IP flows				✓					✓
IDS overview	–		✓				✓			✓
Detection techniques, methods and systems	Statistical	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Clustering		✓			✓	✓		✓	✓
	Classification		✓	✓			✓		✓	✓
	Finite state machines			✓			✓			✓
	Information theory		✓				✓	✓		✓
	Evolutionary computation						✓			✓
	Hybrid/others						✓			✓

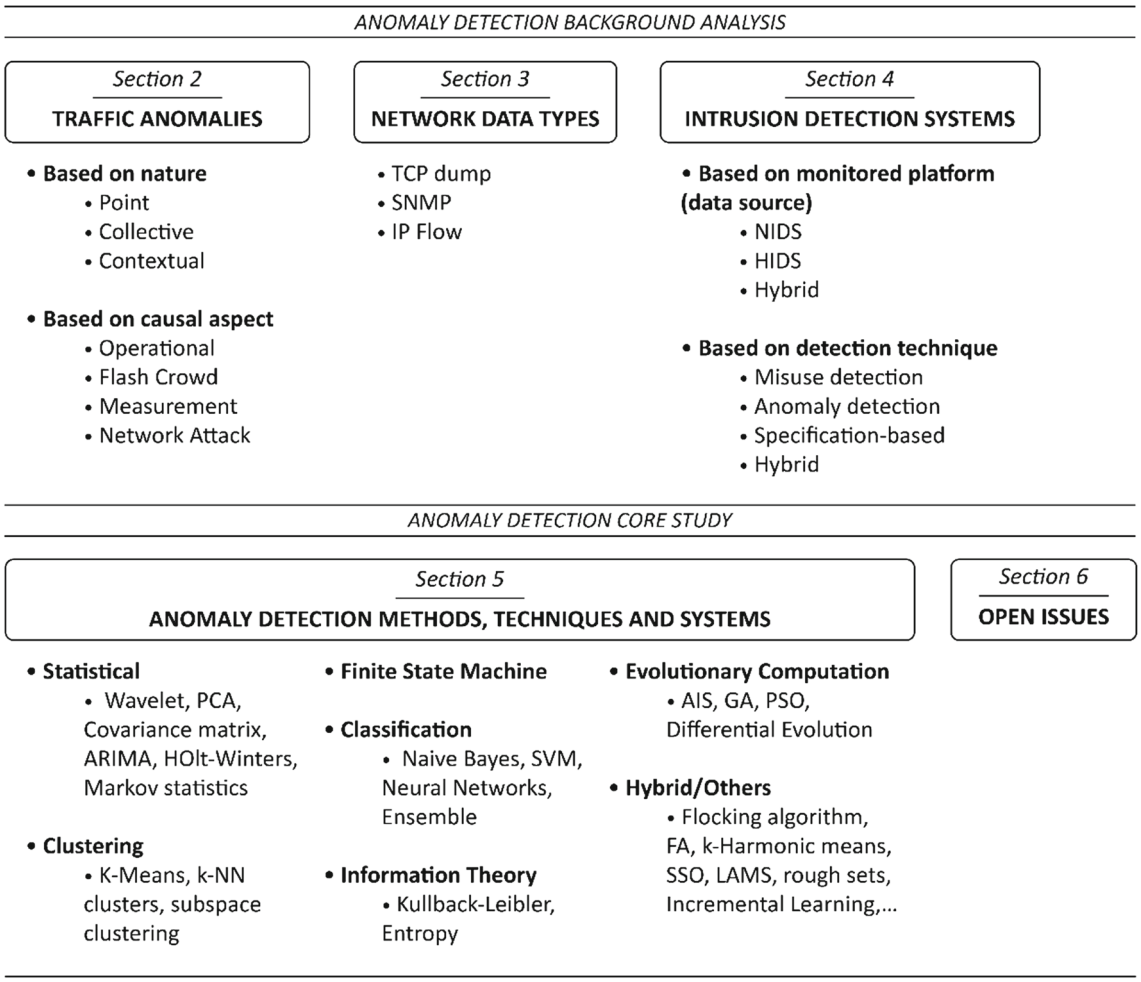


Fig. 1 Paper summary

ing and studying the anomaly context and its categorization, network anomalies can be categorized giving two relevant properties: according to their nature (grouped by how they are characterized, regardless of whether they are malicious or not); and according to their causal aspect (distinguished depending on their cause, regarding either their malicious or non-malicious aspect). Figures 2 and 3 illustrates this categorization and all points that are covered in this section.

2.1 Anomaly categorization based on its nature

The nature of an anomaly is an important aspect of an anomaly detection technique. Depending on the context within which an abnormality is found, or on how it occurred, it can be or not be an abnormality. This aspect can direct how the system will handle and understand mined and detected anomalies. Based on their nature, there are three categories of anomalies: *point anomalies*, *collective anomalies*, and *contextual anomalies* [9,17,18].

A point anomaly is the deviation of an individual data instance from the usual pattern/behavior. These anomalies are the simplest ones, and because of that, they are the focus of most researchers. For better understanding, suppose that the daily spending of a person is one hundred dollars; then, on a specific day, they spend three hundred dollars. This situation characterizes a point anomaly [9,18].

A collective anomaly occurs when only a collection of similar data instances behaves anomalously with reference to the whole dataset. In a collective anomaly, individual anomalous behaviors themselves are not considered anomalies; however, their collective occurrence is considered an anomaly. A point anomaly occurring continuously for an extended period or in a cluster amid background data is a collective anomaly. Consider this example: in a sequence of actions in a computer like "…HTTP-web, buffer-overflow, HTTP-web, HTTP-web, FTP, HTTP-web, SSH, HTTP-web, SSH, buffer-overflow…", the underlined sequence is a collective anomaly. The individual events occurring in other positions in the sequence are not anomalies; however, the

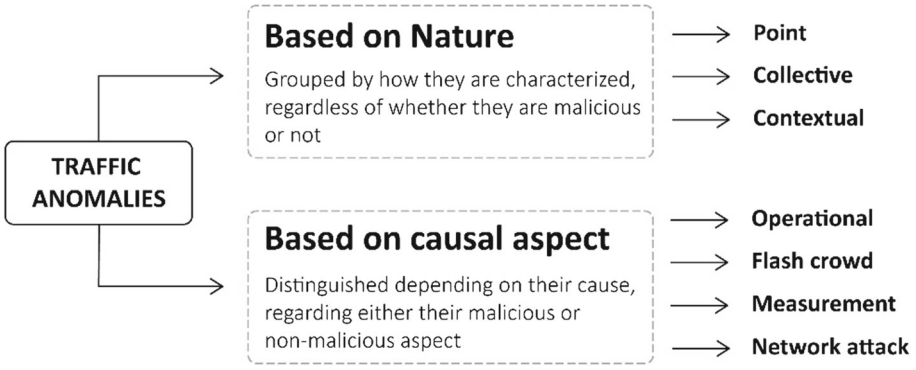


Fig. 2 Traffic anomalies categorization

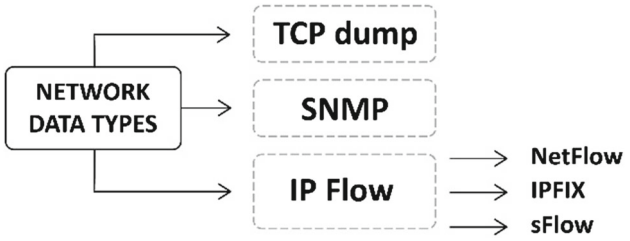


Fig. 3 Network data types categorization

underlined sequence matches a web-based attack by a remote machine followed by the copying of data from the host computer to a remote destination via FTP. Another common example is the ECG exam output, in which low values observed over a long period indicate an anomaly, while one unique low value is not considered abnormal [17,18].

Contextual Anomalies, also called conditional anomalies, are events considered as anomalous depending on the context in which they are found. Two sets of attributes define a context (or the condition) for being an anomaly, both of which must be specified during problem formulation. Contextual attributes define the context (or environment); for instance, geographic coordinates in spatial data or time in time-series data specifies the location or position of an instance, respectively. On the other hand, behavioral attributes denote the non-contextual features of an instance, i.e., indicators determining whether or not an instance is anomalous in the context [9,18,19]. Consider a time-series data set describing the average bits/s of network traffic in a set of days (contextual attribute), in which every day, at 0 h, the server does a regular backup (behavioral attribute). Although the backup generates an outlier in the traffic series, it may not be anomalous since it is normal behavior due to a regular backup. However, a similar traffic outlier at 12 h could be considered a contextual anomaly.

2.2 Anomaly categorization based on its causal aspect

The causal aspect distinguishes anomalies depending on their cause, regarding either their malicious or non-malicious aspect. Anomalies are not always related to attacks intended to harm computer systems or steal information. They can be both events caused by human/hardware failure, bugs or private users when demanding heavy traffic usage, for instance. Thus, as found in Barford et al. [20] and Marnerides et al. [16], anomalies are grouped into four categories: operational/misconfiguration/failure events; flash crowd/legitimate but abnormal use; measurement anomalies; and network abuse anomalies/malicious attacks (or simply, network attacks) [20,21].

Operational events (also called Misconfiguration events or Failures) are non-malicious issues, which may occur in a network system mostly by hardware failures, software bugs or human mistakes. Server crashes, power outages, configurations errors, traffic congestion, non-malicious large file transfers, inadequate resource configuration, or significant changes in network behavior caused by imposing rate limits or adding new equipment, are all examples of this category of anomaly [7]. Such problems can be perceived visually by nearly abrupt changes in bit rate, which appear steady but occur at a different level over a time period [21].

Legitimate but not abnormal use is commonly referred to as flash crowds. Flash crowds are large floods in traffic, which occur when rapid growth of users attempts to access a specific network resource, causing a dramatic surge in server load. Anomalies in this category consist of legitimate requests, which are usually an aftermath of mutual reaction to hot events but far bigger than the load which the system can handle. Flash crowds may occur when a contest result is published on a URL, or when an e-commerce website announces a big sale, or even due to software release. Although it is not malicious, if there is not enough time to react and provide the necessary resources to handle overload demand, these flash events can seriously flood or lead to complete web service failure [22,23]. Flash crowd behavior is related to the rapid growth of particular traffic flow types, such as FTP flows, or the gradual fall of a well-known destination over time.

Measurement anomalies are other issues, which are not network infrastructure problems, abnormal usage, or malicious attacks. These anomalies are related to collection

infrastructure problems and problems during data collection. Examples are the loss of flow data caused by router overload, or when there is a collection of infrastructure problems and the UDP NetFlow transport to the collector becomes unreadable.

Network abuse anomalies (or network attacks) are a set of malicious actions aiming to disrupt, deny, degrade or destroy information and services from computer network systems, compromising their integrity, confidentiality or availability. Numerous types and classes of attacks currently existing may vary from simple email spam to intrusion attacks on critical network infrastructures. Worms, malicious resource abuse, bug exploits and unauthorized access are some examples of common computer attacks. According to Ghorbani et al. [24], attackers gain access to a system, or limit the availability of that system through some general approaches. These are:

- *Social Engineering* when an attacker manipulates people to obtain confidential information, making use of hostile persuasion or other interpersonal tactics [25]. Examples are email phishing and email Trojan horses;
- *Masquerading* this is a type of attack in which the attacker uses a fake identity to gain unauthorized access or greater privileges in a system through official access identification. The attacker illegitimately poses or assumes the identity of another legitimate user, generally by using stolen IDs and passwords [26].
- *Implementation Vulnerabilities* these are cases in which the attacker exploits software bugs in their targets, such as software, services or applications, in order to gain unauthorized access. Examples are the buffer overflow vulnerability or the mishandling of temporary files.
- *Abuse of Functionality* malicious activities performed by attackers excessively performing a legal action in order to congest a link or cause a system to fail. A denial-of-service performed on a web-login system by flooding it with valid usernames and arbitrary passwords in order to lock out authentic users, when the allowed login retry limit is exceeded, constitutes an abuse of functionality.

Based on those general approaches of network abuse anomalies (network attacks), there are various classes of attacks. Table 2 shows the main attack, which commonly harms computer networks and is the major target of anomaly detection mechanisms.

3 Network data types

Another essential step required for building an anomaly detection system is choosing the network data source. The nature of the selected data set may dictate which types of anomalies the system can detect. One needs to choose a data source correctly depending on what kind of anomalies and IDS approaches are intended as the focus of the research. Because of that, accurate data characterization results in the better performance of the anomaly detection system. This section presents some of the most popular sources used in the anomaly detection subject.

3.1 TCP dump

Tcpdump is a packet analyzer tool used to monitor packets on a computer network. It shows the headers of TCP/IP packets passing through the network interface. It is a tool for network packet capturing and analysis and is recommended to professionals who need to perform monitoring and maintenance on computer networks, as well as to students who want to understand the operation of the TCP/IP protocol stack. Nevertheless, this type of data is not used as much nowadays due to its limited information.

3.2 SNMP

The Simple Network Management Protocol (SNMP) [34] is one of the widely used standards for managing IP network components. This protocol has a client-server structure (SNMP managers and SNMP agents) which runs throughout the UDP protocol [35]. SNMP data has been used on intrusion detection systems, since it is useful when it comes to collecting accurate network activity data at a single host level. All collected data are stored, as SNMP objects, in a hierarchical database called MIB (Management Information Base). SNMP objects are summary traffic data constructed by the aggregation of raw data (pcap records) collected mostly by TCP dump tools [16].

Although efficient in their proposals, the works by Cabrera et al. [36] and Yu et al. [37] are limited to detecting only DoS/DDoS attacks, since these are volume anomalies and SNMP objects rely on volume attributes (bits and packet counts). As presented in Moises et al. [38] and Zarpelao et al. [39], the proposed alarm systems developed over SNMP data have shown high anomaly detection rates by combining clustering and parameterizing techniques. However, none of them had any other information about unknown anomalies, despite the alarms being triggered.

A significant advantage is that SNMP is still a widely deployed protocol with available fine-grained data. It is used in traditional network management tools for measuring performance parameters such as error counter interfaces and traffic volume. Packet and bit interface counters are useful; however, nowadays, understanding which IP addresses are the source and destination of traffic and which TCP/UDP ports are generating traffic is vital.

Table 2 Detailed description of most common network abuse anomalies

Attack	Definition	Examples	T[a]
Virus	Piece of code inserted into a file or program which replicates itself without the user's permission	Rootkit.Sirefef.Gen, Trivial.88.D	
	Harmful activities: theft of private information, data corruption, spam messages		
	Needs human intervention to abet its propagation [27]		
Worm	Self-replicating software designed to spread through the network	Morris, CodeRed, Nimda	
	Exploit security or policy flaws in widely used services [28]		
Trojan	A piece of program masquerading as a benign application, when in fact it secretly malicious activities	ZeroAccess Rootki, Beast, Zeus	SE
	They do not replicate as viruses and worms do but can be just as destructive		
Buffer overflow	Takes over programs through buffer vulnerabilities to execute arbitrary code in order to store more data in a buffer than the buffer can hold	–	IV
	Can corrupt or overwrite valid data held in a buffer		
Denial of service (DoS)	Malicious attempts to deny access to shared network resources or services	SYN flood, HTTP flood, ping of death (PoD), RUDY, teardrop, Slowloris	AF
	Generally, it uses significant packet volume containing useless traffic to congest and waste resources serving legitimate traffic. It can be a single or multi-source attack [29]		

Table 2 continued

Attack	Definition	Examples	T[a]
Distributed DoS (DDoS)	DDoS are DoS attacks; they are easy to launch and difficult to locate their source since they are implemented by a group of computers (botnet)	UDP flood, TCP flood, Slowloris, Zero-day DDoS, NTP amplification	AF
	Defeat the target server while keeping their identity unknown by using compromised computers		
Distributed reflective DoS (DRDoS)	Attacks that just cannot be addressed by traditional on-premise solutions. These use legitimate hosts (reflectors) to flood a large number of response packets to the target system by using spoofed IP addresses	Smurf attack, Fraggle attack	AF
	The attacker sends many requests with a spoofed source IP address (the target server address) to legitimate nodes (reflectors), which reply with several voluminous responses to the spoofed IP (target server), thus flooding the victim [30]		
Stealthy attack	Quietly introduced and remain undetected by hiding the evidence of the attacker's actions	Stealthy packet dropping	
Physical attack	An endeavor to harm physical components of a computer or network	Cold Boot attack, Stoned Boot, Evil Maid	AF
	Attackers with physical access to a computer can retrieve encryption keys from a running operating system, for instance		
	As soon as a computer is physically controlled, it can be destructive [31]		
Password attack	Attempts to gain passwords	Dictionary attack, phishing attack	IV
	They are specified by a series of unsuccessful logins (brute force) in a short period of time [32]		

Table 2 continued

Attack	Definition	Examples	T[a]
Cyber reconnaissance	Information gathering attack Gathers information on network systems and services Exploits vulnerabilities or weaknesses by scanning or probing devices or systems [33]	Ping sweeps, Port scans, packet sniffers	IV
Probe	It is accomplished before an attacker launches an attack on a given target Scans or probes the target's network or host by searching for vulnerabilities, open ports, valid IP addresses, services offered, operating system used, etc	IPsweep, portsweep	IV
User to Root (U2R)	Consists of unauthorized access to local superuser privileges by starting as a regular unprivileged user U2R attacks may end in substantial loss of time and money	Loadmore, perl, Xterm	M
Remote-to-Local (R2L)	Unauthorized access via a remote machine Remote to local attack detection using a supervised neural network	FTP write, Warezmaster	M

[a]Type, Network attack type; SE, Social Engineering; M, Masquerading; IV, Implementation Vulnerabilities; AF, Abuse of Functionalities

3.3 IP flow

IP flow analysis is a complete management technology that has been used as an alternative to the SNMP protocol. The development of new services and the increasing complexity of networks led to a need for more detailed information on transmitted data, which is essential in understanding application behavior, users, business departments and other structures relying on the network for their operation.

Accordingly, using flow management tools and protocols allows the construction of a detailed database composed of essential traffic information, enabling the better understanding of more subjective aspects of network operation [40,41]. Thus, it was necessary to go beyond the limited bit and packet counters provided by SNMP in order to characterize more specific traits in the traffic, showing network trends and behavior. Moreover, although packet and byte interface counters are useful, knowing the source and destination IP addresses of the traffic, and which applications are producing it, is invaluable [42].

As a result of these constraints, Cisco Systems presented the NetFlow protocol in 1996 [41,43] and pioneered the introduction of flow structure. A flow [44] is defined as a set of IP packets passing through an observation point over a predefined time interval. All packets constituting a flow have a set of common properties including source/destination IP addresses and TCP/UDP ports, VLAN, application protocol type (layer 3 from the OSI model) and TOS (Type of Service). Moreover, a flow also has some other important attributes, such as byte and packet counts, timestamps, class of service (CoS) and router/switch interface. NetFlow introduced a new practice to assist network management. This was the NetFlow probe, embedded into the network devices (switches), which captures all packets coming through the switch and aggregates them into IP flows according to their common properties. Then, after the timeout of the previously established maximum flow duration, flows were exported out to a collector responsible for analyzing the flow data [45]. NfSen [46] and nTop [47] are the most common graphical applications enabling the analysis of exported flow data.

Besides NetFlow, there are other protocols that have emerged for the same purpose. sFlow was introduced by the InMon Corp. in 2001 [48,49]. Its major difference to other protocols is the usage of random sampling mechanisms during traffic flow aggregation. This feature is appropriate for high-speed networks (gigabit or more). By the year 2008, the Internet Engineering Task Force (IETF) standardized the export of IP flow information from routers, probes, and switches by introducing the IPFIX (IP Flow Information Export) protocol [44]. IPFIX was based on NetFlow version 9; it was developed with more flexible data handling and is able to operate regardless of which transport protocol or message formats are used. Recently, two NetFlow enhancements appeared. Flexible NetFlow uses an extensible format and can export other features apart from the traditional ones. It also has the immediate cache concept, which lets the direct export of flow information without hosting a local cache. NetFlow-lite [50,51] comes at a lower price tier, compared to standard NetFlow, due to not using expensive customer application specific integrated circuits (ASIC). It offers flexibility, similar network visibility and maintains the same packet forwarding performance.

There are several advantages of using flow traffic to detect anomalies [52–54]:

- Lower processing cost. Since flow-based IDSs are based only on packet headers, they only process a small number of flows compared to the big amount of packets processed in packet-based approaches
- Reduced privacy issues, such as the packet's payload, are not considered in the analysis
- Detailed traffic data, mainly regarding NetFlow v9 and IPFIX

Regarding the disadvantages of developing anomaly detection methods under IP flow data, most of them rely on the following:

- Untrustworthy state of UDP protocol and drawbacks of SCTP in confronting scenarios, where multiple network interfaces (routers and switches) need to interact with multiple NetFlow data collectors.
- There is also difficulty in understanding end-to-end traffic, since it may be passing through many hops and routing paths and changing dynamically.
- Although sampling techniques for both flow and packets are efficient in reducing the load of exported and aggregated traffic, respectively, they offer a non-reliable view of the entire network operation. Many researchers have discussed the problems and proposed solutions to optimizing sampling mechanisms; namely, Bartos et al. [55], Zhang et al. [56] and Silva et al. [57].

Table 3 compares the two data sources discussed in this section.

4 Intrusion detection systems

Intrusion Detection Systems (IDS) are automated defense and security systems for monitoring, detecting and analyzing hostile activities within a network or a host. Although the name "Intrusion detection" suggests that these systems actually detect "intrusions", it is not that simple. Kemmerer and Vigna [58] say that, in fact, IDSs do not detect intrusions at

Table 3 Comparison between commonly used data sources for network anomaly detection

Source	Advantages	Disadvantages
TCP dump	Provides comprehensive information about the operation of the TCP/IP protocol stack	Limited information
SNMP	Widely deployed protocol	Only packet and bit interface counters
	Available fine-grained data	No IP nor TCP/UDP ports information
IP Flow	Lower processing cost	Untrustworthy state of UDP protocol
	Based only on packet headers	Drawbacks of SCTP in confronting scenarios where multiple network interfaces need to interact with multiple flow data collectors
	Reduced privacy issues	Difficult to understand end-to-end traffic
	Detailed traffic data	Sampling techniques offer a non-reliable view of the entire network

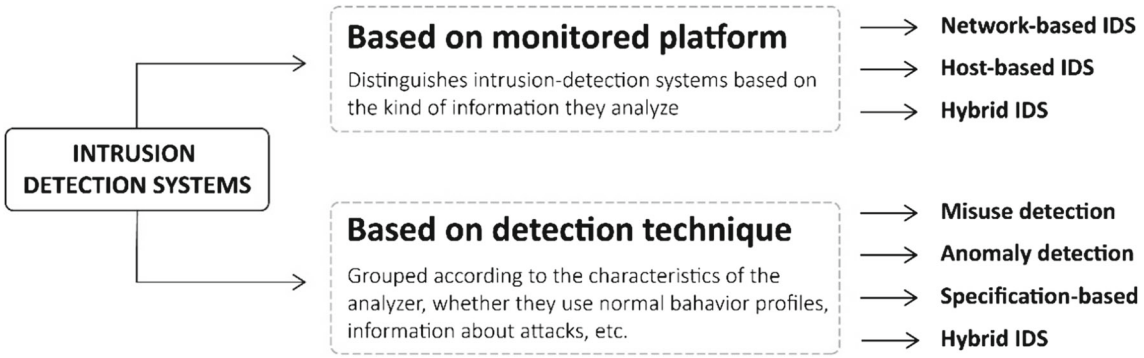


Fig. 4 Intrusion detection systems categorization

all, but they are only able to recognize evidence of intrusions, either during or after the circumstance.

Additionally, Lee and Stolfo [59] state that there are four essential elements to be considered when creating an IDS: resources to protect (accounts or file systems, for instance); models to identify the typical behavior of these resources; techniques that compare the actual activities of these resources with their normal behaviors; and finally, identify what is considered abnormal or intrusive. Apart from these basic IDS functions, they may also be able to provide reports for network administrators and track user policy violations as well as to take self-measures to stop threats or correct problems [13,17,60].

An IDS detects hostile activities by either monitoring network traffic, gathering packets (mostly as a kind of sniffer) to analyze possible incidents, or by analyzing computational system events (such as log files, for instance), in search of security policy violations, unusual use, etc. These incidents may occur due to various reasons, from malware (worms, spyware, etc.) to unauthorized access attacks. The goal of any IDS is to guarantee the security of a network or computer system with regard to confidentiality, integrity, and availability. A firewall is commonly the first defensive line in a network and an IDS is used when there is evidence of an intrusion/attack, which the firewall was unable to stop or mitigate. The IDS then works as the second line of defense. Furthermore, the task is difficult, and in fact intrusion detection systems do not detect intrusions at all, they only identify evidence of intrusions, either while they are in progress or after the fact

IDSs can be categorized in many ways [61]. Depending on the monitored platform (data source), IDSs are divided

into three types: network-based IDS (NIDS), host-based IDS (HIDS), and hybrid. Furthermore, regarding the technique of detecting unusual activity, IDSs can be categorized into four types: anomaly-based IDS, signature-based IDS, specification-based IDS, and hybrid. Figure 4 and Table 4 condense the seven aforementioned IDS types, which are presented and discussed thoroughly in subsequent sections.

4.1 IDS types by monitored platform (data source)

4.1.1 Network-based IDS (NIDS)

A network-based IDS is deployed in order to detect intrusions in network data over network connections and to protect all network nodes. Since intrusions usually occur as irregular patterns, this kind of IDS analyzes and models traffic to identify the occurrence of regular traffic and suspicious activities. They are composed of a set of sensors placed at many network points in order to monitor traffic. Each sensor performs a local analysis and reports suspicious activity to a central management console. A network-based IDS is capable of gathering and analyzing entire transmitted packets as well as their payloads, IP addresses, and ports.

NIDS are effective for monitoring both inbound and outbound network traffic. This type of IDS ensures that a large network can be monitored with only a few installed IDSs, as long as they are well positioned. It is usually simple to add this type of IDS to a network and they are considered well secured against attacks. However, they have some disadvantages, such as the difficulty in processing all packages from a large and overloaded network. Thus, they may fail to recognize an attack launched during periods of intense traffic. Moreover, many of the advantages of network-based IDSs do not apply to more modern networks based on *switches* since they segment the network and require enabling monitoring ports for the sensors to function properly. Port mirroring or spanning is used to enable a complete view in a switched network; however, this causes overhead.

Another disadvantage of network-based IDSs is that they are unable to analyze encrypted network packets, since those appear only on the target machine. Finally, since NIDSs can detect the presence of suspicious activities, there is no reassurance for their success or failure [17,60,62]. Figure 5 illustrates a conventional network-based IDS.

4.1.2 Host-based IDS (HIDS)

A Host-based IDS is set to operate on specific hosts (single PCs). Its focus is to monitor events on the host and detect local suspicious activities, i.e., attacks performed by users of the monitored machine or attacks occurring against the host where it operates.

Since this type of IDS is designed to operate with only a host, it is capable of specific tasks, which are not possible with an NIDS, such as integrating code analysis, detecting buffer overflows, monitoring system calls, privilege misuse, privilege abuse, system log analysis, and others. These systems are classified as agent-based, since they require the installation of software on the host. This IDS evaluates the safety of the host based on operating system log files, access log, and application log, for instance. It is vital because it provides security against the types of attack that the firewall and NIDS do not detect, such as those based on encrypted protocols, since they are located at the destination. Another benefit of HIDS over NIDS is that the success or failure of an attack can be promptly determined [17,60,62]. Figure 6 illustrates a general host-based IDS.

4.1.3 Hybrid IDS

Hybrid IDSs are developed considering data provided by the host events and the network segments and by combining the functionalities of both network and host-based IDSs [61]. These systems aggregate the benefits of both approaches while overcoming many of the drawbacks. However, hybrid systems may not always mean better systems. Since different IDS technologies analyze traffic and look for intrusive activity in various ways, getting these different technologies to interoperate and coexist in a single system successfully and efficiently is a challenging task.

4.2 IDS types by detection technique

4.2.1 Signature-based (misuse detection)

Signature-based techniques, also known as knowledge-based or misuse detection, evaluate network activities by using a set of well-known signatures or patterns of attack stored in the IDS database. Whenever an attempt matches a signature, the IDS triggers an alarm. This operation ensures an efficient detection with minimal false alarms, and a good level of accuracy with regard to the identification and classification of abnormalities, making it easier for network administrators to take preventive or corrective measures.

However, as any other action not recognized by the IDS knowledge database is considered normal, unknown anomalies, or little variations in known attacks, cannot be detected. For this reason, signature-based IDSs require constant updating of their knowledge database. Signatures must be defined in order to ensure that all probable variations of an attack are covered. Additionally, they do not match non-malicious activities, which can be a hard task [16,24,63,64]. Generally, misuse detection techniques work as shown in Fig. 7.

Table 4 IDS type classification and organization summary

Classification	IDS type/description	Advantages	Disadvantages
Data source/monitored platform	*Network-based (NIDS)*	Monitor both inbound and outbound network traffic	Difficulty in processing all packages from a large and overloaded network
		Detect network-specific attacks, such as denial-of-service	Failure to recognize attacks launched during periods of intense traffic
		Detect known worms and viruses, flash crowds, port scan	Unable to analyze encrypted packets
			Demand for more sensors in today's large networks is costly
	Host-based (HIDS)	Detect local suspicious activities	Incomplete network picture
		Detect attacks based on encrypted data, since they are located on the destination	Since they are agent-based, support for different operating systems is required
		Privilege abuse, buffer overflows	
	Hybrid	Aggregate benefits of both approaches	Get distinct approaches to interoperate and coexist in a single system
		Overcome many drawbacks	
Detection technique	*Misuse detection*	High detection accuracy	Unable to detect unknown anomalies
	Use of prior-knowledge attack database (signatures)	Low false alarm rate	Difficult and time-consuming task to build and update signatures
	Anomaly detection	Detect both known and unknown anomalies	High false positives and false negatives
	Profile representing normal network behavior	Discover new attacks (and use on signature-based IDSs)	Less efficient in dynamic network environments
		No demand for prior knowledge	Demand time and resources to construct the profile
	Specification-based	Unknown attacks discovery	Complexity
	Set of constraints to describe and monitor the operation of a program or protocol	Low false positive rates	Elaboration of detailed specifications and constraints is costly and time consuming
		Resistant to subtle attack changes	Restricted to the proper operation of a program or protocol
	Hybrid	Aggregate benefits of the three approaches	Get distinct approaches to interoperate and coexist in a single system
		Overcome many drawbacks	

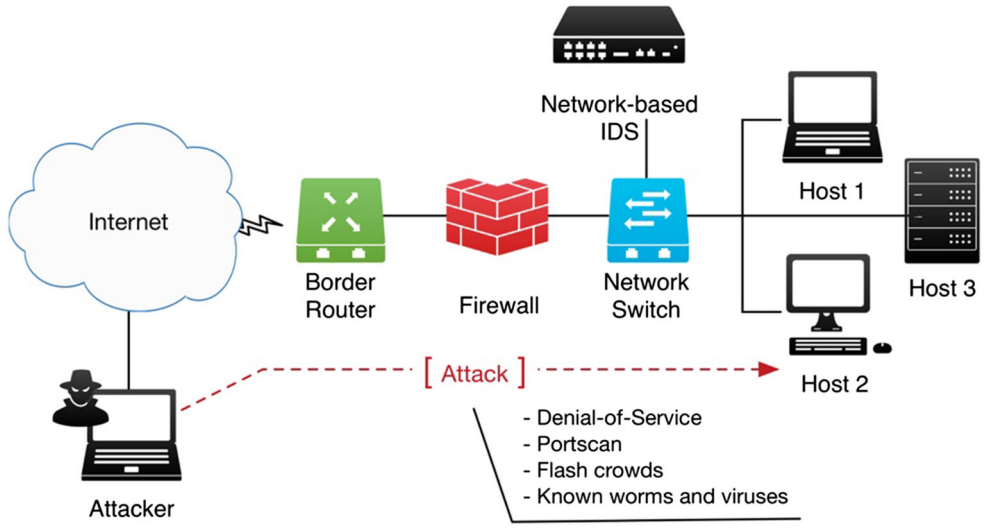


Fig. 5 Network-based IDS example

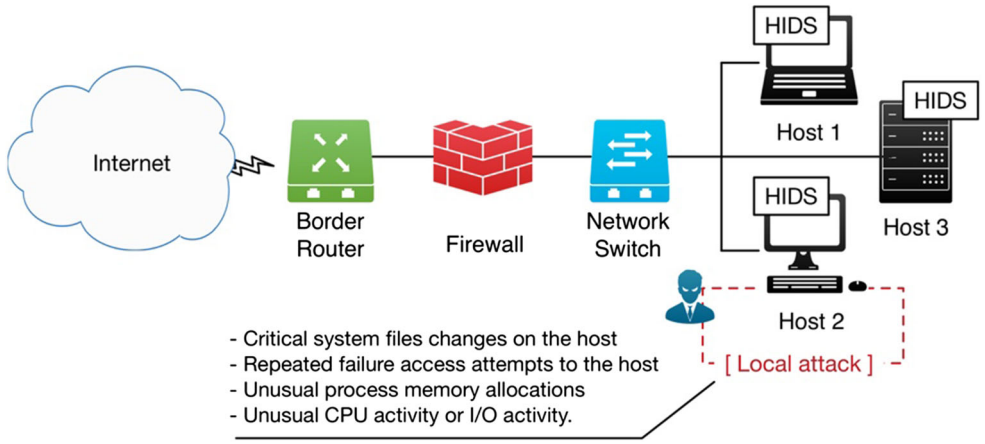


Fig. 6 Host-based IDS example

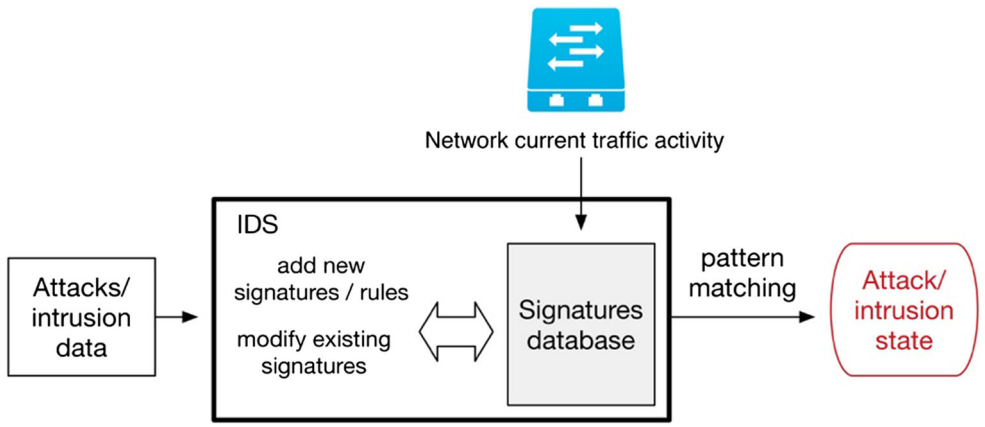


Fig. 7 Misuse detection (signature-based) techniques general scheme

4.2.2 Anomaly-based (anomaly detection)

Anomaly-based techniques, also known as profile-based or anomaly detection, are founded on the creation of a baseline profile representing normal/expected network behavior, and on that any observed deviation of current activity compared to this profile is considered anomalous. This profile is generated mostly through statistical and historical network traffic data.

A classic example of this type of detection is when a specific user always uses the Internet for a certain period of the day, during business hours. Imagine that this user is a manager at a company being monitored by an anomaly-based IDS. This IDS has spent a whole week creating this user's normal profile, and from the last day of that week, it employed this profile as mandatory for the time allowed to use the Internet. While detection is active, the manager wants to use the Internet during night-time in order to submit a last-minute report, which is something unusual to regular usage. The response of the anomaly-based IDS to this unusual behavior is to restrict Internet access to that user, which would be valid if this was not an exception; however, this would actually be treated as a false positive.

Therefore, the main drawback of profile-based techniques is the possibility of increased false alarm (false positive) rates, because users and system behavior may widely vary. Additionally, attacks may be launched during the learning period and result in a profile containing intrusion behavior, which may not be able to detect some anomalous behaviors. These are false negatives, which is even more serious than false positives. Therefore, constant retraining of the profile is required; however, this may cause the unavailability of the detection system or an increase of false alarms [65]. Finally, depending on the approach, profile creation may demand an extended monitoring period or high computational resource usage.

Anomaly detection techniques are the most commonly used IDS detection type. This is due to their ability of detecting both known and unknown attacks and anomalies, since the detection is performed under the discovery of unusual patterns, which makes this technique more dynamic than the static signature-based technique. It is also helpful in discovering new types of attack and behavior, and as a knowledge builder for new signatures in misuse detection systems.

Anomaly-based detection is the most popular and well-investigated topic among researchers. There are many different techniques and algorithms, described in the literature, used to build this normal profile and find unusual patterns, such as statistical procedures, machine learning, clustering, fuzzy logic, and heuristics. This has been studied for over 20 years, and there is still a wide investigation panel to be discovered, as well as critical challenges and open issues to overcome, as will be presented later in this survey. Figure 8 shows the general structure of an anomaly detection approach.

4.2.3 Specification-based

As described in [63,66], anomaly detection systems detect the effect of abnormal behavior, while misuse detection systems recognize already known abnormal behavior. Accordingly, specification-based methodologies were created in order to utilize the benefits of both techniques. Therefore, these IDSs manually develop specifications and constraints to characterize normal network behavior. This methodology is accomplished by obtaining the correct operation of a program or protocol and monitoring its execution through the definition of set constraints. Accordingly, this methodology can be more resistant to suitable changes in attacks and allows the discovery of previously unknown attacks while having a very low false positive rate.

On the other hand, specification-based techniques are much more complex since their analysis can be performed in the layers existing below the application layer of the Internet protocol stack, or at the operating system control level. These techniques are restricted to the proper operation of a program, or protocol, and can be excessively tedious and susceptible to errors since they rely on user knowledge. Furthermore, the elaboration of detailed specifications and constraints is costly and time consuming.

This detection model is not as widely distributed as others cited in this paper, especially because of its greater development complexity and restricting the intended application, since it is aimed, for example, to be a single application.

4.2.4 Hybrid techniques

Hybrid IDSs, or Compound detection, implement combinations of misuse, anomaly and specification detection techniques. These systems can be based on the normal network profile and also attack behavior, for instance.

An example of a hybrid IDS has been proposed by Assis et al. [67], in which a network profile called digital signature of network segment using flow analysis (DSNSF) was created to detect unknown anomalies within network traffic. Then, pre-loaded signatures classified the discovered anomalous behavior as a DoS, DDoS, flash crowd or port scan attack. Another example has been presented by Stakhanova et al. [68], who combined specification-based techniques with anomaly-based ones in an effort to mitigate the limitations of the former. The need of user expertise is overcome by an approach for the automatic generation of normal and abnormal behavioral specifications as variable-length patterns, which are classified via anomaly-based machine learning techniques.

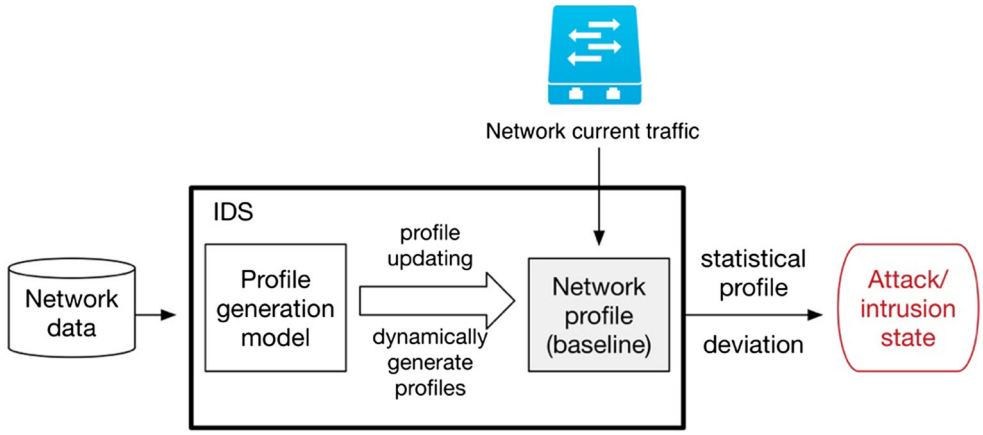


Fig. 8 General scheme of anomaly detection (anomaly-based) techniques

5 Anomaly detection techniques, methods and systems

In this survey, we focus on anomaly detection (anomaly-based IDS); the following chapters contain a review of its most current techniques, methods, and systems. However, since there are many emerging types of research proposing hybrid approaches, the combination of both misuse and anomaly detections, for instance, may be addressed as well (Fig. 9).

5.1 Statistical methods

Statistical methods for anomaly detection are widely used and are commonly based on probabilistic models associated with training data for the purpose of tracking network behavior. Anomalies are related to sudden changes in network data. Mostly, these abrupt changes are detected by modeling hard thresholds. The primary challenge for statistical techniques is to find methods reducing false alarm generation caused by hard thresholds [12]. For instance, statistical signal processing procedures may be used to increase the detection rate while decreasing false alarms, as Lakhina et al. did in their work with principal component analysis [10,69,70].

5.1.1 Wavelet analysis

Wavelet analysis focuses on modeling non-stationary data series'. Such data series may contain signals that can vary in both amplitude and frequency over extended periods of time. Unlike Fourier analysis, which uses trigonometric polynomials, data series are modeled using wavelets, which are powerful basis functions localized in time and frequency, allowing a close connection between the series being represented and their coefficients. In this manner, wavelet analysis is fundamentally a way to describe levels of detail with regard to particular data, which can be images, curves, surfaces, and so on.

Callegari et al. [71] propose a real-time anomaly detection method using wavelets combined with sketches. It is a router level analysis performed by extracting NetFlow traces and transforming them into ASCII data files. After formatting, sketches are used to aggregate different traffic flows in sketch tables through hash functions. Next, the time series are submitted to a wavelet transform for the purpose of discovering discontinuities.

Another study using wavelets was produced by Hamdi and Boudriga [72]. It relied on identifying attack-related anomalies by differentiating between dangerous and non-threatening anomalies. This task was achieved based on the concept of period observation, where wavelet theory was used to decompose one-dimensional signals in order to analyze both their special frequencies and time localization.

5.1.2 Principal component analysis

Principal component analysis (PCA) is a widely used statistical technique for anomaly detection in computer networks. It is defined as a dimensionality reduction approach, in which a data set consisting of n correlated variables can be mapped onto a new and reduced set of k variables, the principal components (PCs), where $k \ll n$. These PCs are a set of orthonormal vectors, which define a k-subspace, and are uncorrelated and arranged so that the first components retain most of the variation present in all original variables [73,74].

Lakhina et al. [10], who pioneered this field, addressed the anomaly diagnosis problem in network wide-traffic by using PCA to efficiently separate traffic measurements into normal and anomalous subspaces. The main idea was that PCA results in a reduced set of k variables (principal components or k-subspace) which corresponds to normal network traffic behavior, while the remaining subspace of m components

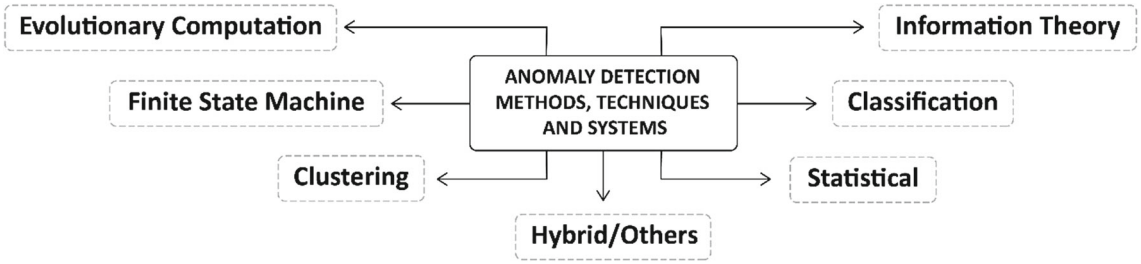


Fig. 9 Anomaly detection methods, techniques and systems analyzed in this research

($m = n - k$) consist of anomalies or noise. Then, every new traffic measurement is projected onto both subspaces so that different thresholds can be set to classify these measurements as normal or anomalous. Their work was responsible for the massive attention on PCA-based approaches for anomaly detection received in the last decade. However, although it was a notorious work with good results and advances in the area, it received some criticism from various authors, mainly related to the calibration sensitivity of PCA, as reported in Ringberg et al. [75].

Ringberg et al. [75] and others [76,77] have criticized the studies of Lakhina et al. [10,70] on PCA by outlining four main challenges regarding its sensitivity: (i) false positive rates are affected by small noises in the normal subspace; (ii) the level of traffic aggregation can mitigate the value of PCA; (iii) large anomalies can infect the normal subspace; (iv) no mapping amongst the reduced subspace PCA produced and the original spatial source of the anomaly.

In this manner, the anomaly detection method proposed by Pascoal et al. [78] used a robust PCA detector merged with a robust feature selection algorithm in order to obtain adaptability to distinct network contexts and circumstances. Additionally, this robust PCA approach does not require perfect ground-truth for training, which is one of the limitations of standard PCA discussed in [75]. In [79], the authors propose ADMIRE, which is a combination of three-step sketches and entropy-based PCA, and results in better true and false positive rates while being capable of capturing distinct kinds of anomalies due to the different entropy time series for PCA. Furthermore, O'Reilly et al. [80] surpassed those limitations in finding the PCs from a dataset with anomalies by proposing a Minimum Volume Elliptical PCA (MVE-PCA) method, consisting of the solution to a convex optimization problem by creating a soft-margin minimum volume ellipse around the training dataset, which decreases the effect of anomalies existing in the data.

Nevertheless, Camacho et al. [81] actively maintain that neither the original PCA proposal nor critical researchers could effectively surpass the disadvantages of using PCA for anomaly detection. To overcome these drawbacks, the authors used a PCA-based multivariate statistical process control (MSPC) approach, which monitors both the Q-statistic and D-statistic. Thereby, it was possible to establish control limits in order to detect anomalies, when they became consistently exceeded. Additionally, the MSPC approach has contribution plots used for finding the root cause of the anomaly. Data pre-processing relies on the feature-as-a-counter approach in which variables are counters for the number of times some event is logged throughout a given time interval. This is in contrast to the idea of Lakhina et al. [10], which considers counters as simple quantitative variables.

Fernandes et al. [82,83] proposed PCADS-AD, an autonomous profile-based anomaly detection system based on a dimensionality reduction procedure and principal component analysis (PCA). It is an enhanced version of their initial work presented in [84]. The system was divided into two main stages. First, the authors used a different interpretation of PCA to generate a network profile called Digital Signature of Network Segment using Flow analysis (DSNSF). The system analyzed historical network traffic data over a period of days, identifying among them the most significant traffic time intervals while reducing the data set so that the new reduced set could efficiently characterize normal network behavior. Then, the DSNSF was used as a threshold to detect volume anomalies by restricting an interval, where deviations were considered normal, through some PCA parameters. This system used three IP flow features (bits/s, packets/s and flows/s) to predict network normal behavior and generate the DSNSF. Another four flow attributes (origin and destination IP addresses and TCP/UDP Ports) were used to produce a report containing useful information concerning the abnormal traffic interval; thus, the network administrator was assisted in taking fast measures to resolve the identified problem. The drawback of this approach is the usage of only volume attributes for anomaly detection, which only considers the detection of volume-based attacks. In this manner, the system is unable to detect attacks which do not impact on bits, packets, and flows.

5.1.3 Covariance matrix

Covariance matrices are second-order statistics and have been proven to be a powerful anomaly detection method. An interesting direction in this area is finding which vari-

ables best label network anomalies and improve detection performance.

The work presented in [85] employs covariance matrix analysis to detect flooding attacks. This approach models network traffic as covariance-matrix samples in order to make use of statistical assets contained in the temporally sequential samples for the purpose of detecting flooding attacks. Then, it directly uses changes of covariance matrices and differences of correlation features to reveal the alterations between normal traffic and various types of flooding attacks.

Miao Xie [86] performed anomaly detection in a segment-based manner by handling a collection of neighboring data segments, with the aid of random variables, and exploiting their spatial predictabilities to determine which ones behaved abnormally. This approach used a sample covariance matrix approximated per the concepts of Spearman's rank correlation coefficient and differential compression in order to substantially reduce the computational cost.

Huang et al. [87] supported the use of covariance matrix for dimensionality reduction instead of traditional PCA discussed in the previous section. They pointed out that a static choice of k principal components is poor at capturing real-time changes, in addition to only allowing weak heuristics due to sensitivity to small variations in the dimensions representing the normal subspace. Therefore, to overcome the limitations of variance-based approaches, the authors came up with a distance-based dimensionality reduction approach for anomaly detection. Depending on their types, anomalies manage to cause distinct types of deviances in the covariance matrix of observed traffic. These deviances allow the categorization of detected anomalies and immediate decision-making with regard to mitigation actions. Their proposal was also able to adapt to changing patterns in the test data such that the model would only use a few important dimensions at any time.

5.1.4 Others

This section presents other noteworthy statistical methods, which do not fit into the previous ones since they combine different statistical techniques.

The study by Kalkan and Alagöz [88] used traffic filtering as a way to prevent network attacks and especially DDoS attacks. *ScoreForCore* was classified as a statistical filtering model based on reaction time and collaboration, which selects the most suitable features from the attack related traffic. The model calculates a score for each packet using the nominal and current profile; then, it compares them in order to find the two features deviating the most from the nominal profile by using collaboration between routers and thresholds. Ozkan et al. [89] studied the anomaly detection problem for fast streaming temporal data, in an online setting, and proposed an efficient statistical online algorithm fusing Markov statistics with Neyman–Pearson (NP) characterization. Their proposal successively learns the feasible varying nominal Markov statistics in a time series and detects anomalous subsequences by first assigning scores to each fixed length subsequence using pair-wise distances and then considering the magnitude of the anomaly score and providing Neyman–Pearson characterization.

Network traffic is currently composed of cycles consisting of bursts with specific characteristics directly affected by working days and user access periods. Under this assumption, Proença et al. [90] introduced the Digital Signature of Network Segment (DSNS), which is a set of information capable of defining the traffic profile. It automates the task of monitoring network segments by statistically estimating the traffic behavior based on historical traffic data. The algorithm is called BLGBA and is based on a variation of the statistical measure *mode*. After extracting SNMP traffic samples from the MIB, the DSNS is built second by second through the analysis of a prior period. The calculation distributes the elements in frequencies according to differences between the size of each sample. Then, the authors validated the DSNS through visual analysis, Bland–Altman plots, residual analysis, linear regression, and the Hurst parameter.

A correlational paraconsistent machine (CPM) has been proposed by Pena et al. [91] and relies on two unsupervised traffic characterization methods and non-classical paraconsistent logic (PL). The authors used both ant colony optimization for digital signature (ACODS) and auto regressive integrated moving average (ARIMA) [92] methods in order to analyze historical network traffic data and generate two distinct network profiles able to describe normal traffic behavior. These profiles are called digital signature of network segment using flow analysis (DSNSF) and is derived from the work proposed in [90]. The existence of anomalies is related to degrees of certainties and contradictions produced by paraconsistent logic over a correlation between two prediction profiles and associated real traffic measurements. From the Euclidian distance calculation between the two DSNSFs and the evaluation of paraconsistent logic signals, the model obtains real evidence for the proposition P (P -> "interval contains an anomaly") to be true.

Another statistical traffic characterization approach for anomaly detection by creating the DSNSF network profile is proposed by Assis et al. [67]. It is a seven-dimensional profile-based anomaly detection system based on the Holt–Winters forecasting technique. The IP flow traces bits/s, packets/s, flows/s, origin and destination IP addresses, and Ports, are simultaneously analyzed in every one-minute time window; therefore, the system can identify different anomalies and generate alarms. The normal network profile of how the network should behave in the next day is predicted dynamically by using the current traffic of the day and the previous day's generated profile. Authors use thresholds to

indicate the interval between real traffic and the profile considered as normal. These thresholds are calculated in an asymmetric way, using the profile, a scaling factor for its width, and a deviation measure. The intervals with mostly greater errors are updated with the absolute deviation of the interval while the opposite confidence band is updated with the standard deviation of the profile. Finally, the alarm system is capable of detecting anomalies in two ways: (i) by alerts, which are related to anomalous behaviors not existing in the system anomaly database; and (ii) by alarms, generated when the system knows the anomalous behavior signature.

Bang et al. [93] propose an IDS using a hidden semi-Markov model (HsMM) aimed specifically at the detection of advanced LTE signaling attacks on WSNs. According to the authors, traditional hidden Markov Models (HMM) cannot represent many possible transition behaviors; therefore, HsMM overcomes this limitation since it has arbitral state sojourn time and is more suitable to time-series behavior analysis. They used the HsMM to effectively model the spatial-temporal characteristic of the wake-up packet generation process, taking the process log-likelihood as the test basis of normality. Then, their detector compared observed spatiotemporal features of a server's wake-up packet generation, with the normal criteria established by the HsMM. Therefore, an alarm is set off whenever significant divergence occurs.

Although classical Markov chain techniques are widely accepted in anomaly detection applications, their short memory property may ignore interactions among the data. On the contrary, the long memory property of a higher order Markov model clouds the relationship between previous and current test data and, thus, it reduces reliability. In light of this, Ren et al. [94] defended that once Markov models are established in the training phase, their order is fixed to detect anomalies in the testing phase. However, the fixed Markov models (n-order) force each state of a sequence to be conditioned on previous n states and may not be enough to provide an accurate estimate of the detecting state. Thus, the authors proposed a dynamic Markov model to balance the length of the memory property of Markov models and keep the strong correlation between memory (or the Markov model) and current test data. To achieve this, the proposed approaches repeatedly calculate the Pearson correlation in order to find the proper order of the Markov model in a sliding window, where the sequential data is segmented. To keep detection continuous, a substitution strategy of anomalies was reported to protect the building of models from the infection of detected anomalies.

Jazi et al. [95] explored several types of application-layer DoS attacks and proposed a detection approach based on a nonparametric CUSUM algorithm. The proposed approach relies on a selected combination of application and network-level attributes for anomaly detection. According to the authors, the resulting method was evaluated on various types of attacks on modern web servers since they represent the most common target for DoS attacks. In addition, the study investigated the performance fluctuation in the presence of thirteen different sampling methods and explored the impact of sampling on the detection of application level DoS attacks. The results confirmed that even specialized sampling techniques could introduce some distortion in detection quality. In this manner, detection should be tied to the sampling technique in order to compensate for distortions provided by sampling and to ensure the improved assessment of traffic characteristics.

5.1.5 Summary

In summary, statistical approaches include the following advantages.

- Intrinsic capability to detect network anomalies than any other method,
- Ability to learn the expected behavior of the traffic (network system),
- Traffic analysis is based on the theory of sudden changes, which sets an alarm whenever a significant deviation happens.
- The methods do not require any kind of prior knowledge about the system as an input.

However, there are some relevant drawbacks that must be considered.

- Some kinds of attacks may be a regular part of the training dataset and may be incorporated in the normal behavior, causing it to be considered as normal.
- It requires some relevant time to train the models in order to be able to set the first alarm.
- The use of thresholds may not be reliable in some real-world cases due to its limited and static nature.

Table 5 summarizes the characteristics of discussed statistical approaches, regarding techniques, data precedence, investigated anomalies, and validation metrics used to test detection performance.

5.2 Clustering methods

Clustering analysis aims to group a set of objects into classes of similar objects. These classes, or groups, called a cluster, and its objects, are similar (in one way or another) to each other and dissimilar to those in other clusters. Clustering-based processes are adaptable to changes and help single out useful features distinguishing different groups. Clustering techniques can be used for outlier detection, identifying values, which are too "far away" from any cluster,

Table 5 Comparison of statistical anomaly detection approaches

Paper	Year	Tech.[a]	Anomaly type[b]	Dataset[b]	Source[c]	Validation metrics
Hamdi and Boudriga [72]	2007	Wavelet	DoS/DDoS (S)	Real network from MIT (R)	IP flow	Packet count
Callegari et al. [71]	2011	Wavelet	Generic synthetically added anomalies in the data (S)	Abilene/Internet2 Network (R)	NetFlow	Detection rate
Lakhina et al. [10]	2004	PCA	Synthetic injection of large and small anomalies (S)	Sprint-1 and Abilene backbones (R)	NetFlow	Detection rate, FPR, Identification rate and mean absolute relative error
Pascoal et al. [78]	2012	PCA	Portscans and snapshots (R)	Small private laboratory network scenario (R)	–	Recall, FPR and precision
Kanda et al. [79]	2013	PCA	22 attack categories (TCP SYN flood, port scan, etc.) (R)	Backbone link from the MAWI traffic repository (R)	IP flow	TPR, FPR, accuracy, F-measure, ROC and Euclidean distance
Fernandes et al. [83]	2015	PCA	DoS, DDoS and Flash Crowd (S)	University network (R) and simulated anomalies (S)	NetFlow	NMSE, correlation coefficient, TPR, FPR, ROC
Camacho et al. [81]	2016	PCA	DoS and other general network faults/anomalies (R)	VAST 2012 2nd mini challenge (R) and a controlled scenario (R)	Firewall and IDS logs, NetFlow	TPR, TNR, FPR, FNR, Recall, Specificity, Accuracy
O'Reilly et al. [80]	2016	PCA	Generic (S) (R)	2-dimensional synthetic Gaussian data (S), UCI machine learning repository (R)	–	Area under the ROC curve (AUC), FPR, TPR, ROC,
Yeung et al. [85]	2007	Cov. Matrix	Flooding attacks (DDoS) (R)	KDDCUP 99 (R)	TCP dump	Detection rate, FPR
Xie et al. [86]	2015	Cov. Matrix	Generic (constant, burst, small noise and large noise anomalies) (R) and artificially injected (S)	IBRL network (R)	–	ROC, average saving rate (ASR)
Huang et al. [87]	2016	Cov. Matrix	Generic labeled anomalies (R)	Kyoto2006+ dataset (R)	–	FPR, ROC
Proença et al. [90]	2004	Statistical mode	Generic outliers (R)	University network (R)	SNMP	Hurst parameter, residual analysis, linear regression, Bland–Altman plot
Assis et al. [67]	2014	Holt–Winters	DoS, DDoS, Flash Crowd, portscan (S)	University network (R) and simulated anomalies (S)	NetFlow	Accuracy, TPR, FPR, ROC

Table 5 continued

Paper	Year	Tech.[a]	Anomaly type[b]	Dataset[b]	Source[c]	Validation metrics
Pena et al. [91]	2014	ARIMA, Paraconsistent logic	Generic (R)	University network (R) and simulated anomalies (S)	NetFlow	Real evidence level, TPR, FPR, ROC
Kalkan and Alagöz [88]	2016	Filtering model, reaction time	Different types of DDoS (S)	MAWI Working Group Traffic Archive (R) and simulated environment (S)	–	Precision, recall, TNR, Negative predictive value (NPV), f-measure, f-measure complement, accuracy and attack prevention efficiency (APE)
Ozkan et al. [89]	2016	Markov statistics, Neyman–Pearson	Sudden change in the source statistics (S)	Monte Carlo simulations (S)	–	ROC, FPR
Bang et al. [93]	2017	Hidden semi-Markov model	Advanced LTE signaling attack types (S)	Simulated environment (S)	–	FPR, FNR, TNR
Ren et al. [94]	2017	Dynamic Markov model	Generic outiliers (R) (S)	Synthetic dataset (S) and Shangai airport traffic, UCR archives	–	TPR, FPR
Jazi et al. [95]	2017	CUSUM	Application layer DoS (S)	ISCX dataset, academic network traces (R)	–	Detection rate, FPR

[a] Statistical techniques/methods used
[b] Data precedence; R, Real; S, Simulated
[c] Source types in blank are either not clearly specified by the authors or not relevant in their research

or as a preprocessing step for other algorithms/approaches. Additionally, classification is an effective resource for distinguishing groups or classes of objects; however, it requires the often costly collection and labeling of a large set of training tuples or patterns, which the classifier uses to model each group [96].

Rajasegarar [97] presented a distributed hyperspherical cluster based algorithm for anomaly detection in wireless sensor networks. Clustering was used to model the traffic data at each node by classifying data vectors as either normal or anomalous. Anomalous clusters were identified by using the average inter-cluster distance of the k nearest neighbor (KNN) clusters. This works under a distributed scheme, where sensor nodes report on cluster summaries, which are merged by intermediate nodes before communicating with other nodes and, thus, minimize communication overhead.

Mazel et al. [98] introduce a non-supervised approach to detecting and characterizing network anomalies. This approach initially works by using a clustering technique, combining sub-space clustering with evidence accumulation clustering and inter-clustering results association in order to blindly identify anomalies in traffic flows.

K-means is a popular clustering technique in the anomaly detection field and is able to classify data into distinct categories. However, it has drawbacks such as local convergence and sensitivity to the selection of cluster centroids. Therefore, many researchers try to combine k-means with other techniques in order to overcome these shortcomings. Karami and Guerrero-Zapata [99] introduced a fuzzy anomaly detection system based on the hybridization of particle swarm optimization (PSO) and k-means with local optimization in order to determine the optimal number of clusters. It is divided into two phases: the training phase aims to find the near optimal solution by combining a novel boundary handling approach of PSO's global search with the fast convergence of k-means; thus, it avoids being trapped in a locally optimal solution. The fuzzy approach is used in the detection phase, in which false positive rates are reduced with a reliable detection of intrusive activities. This is due to any data (normal or attack), which may be at close distance to some clusters.

Carvalho et al. [100] developed a proactive network monitoring system that can detect unusual events and reduce manual intervention and error probability in decision-making. Their proposal consists of creating a network profile called DSNSF (digital signature of network segment using flow analysis), which describes normal network usage using a clustering approach through the modification of the ant colony optimization (ACO) metaheuristic, called ACODS. ACODS characterizes network traffic discovery in the large volume of high-dimensional input data in a cluster set, and by optimizing the extraction of behavioral patterns through an unsupervised learning mechanism. Then, to detect anomalous behavior, authors use the pattern matching technique called dynamic time warping (DTW). They first compute the similarity between real traffic and normal profile in each time interval; then, compute the distance between the series and provide a measure based on both form and distance. The proposed alarm system works with seven flow attributes, using entropy to summarize information regarding IP addresses and Port features. When an anomaly is detected, ACODS provides a full report containing IP flow information indicating the impact of each attribute on the detected anomalous time interval. ACODS has a square complexity, resulting in a solution convergence by many iterations, in which authors try to mitigate by using local search and pheromone updating.

In, Dromard et al. [101] proposed ORUNADA, an unsupervised anomaly detector based on the incremental grid clustering algorithm called IDGCA and a discrete time sliding window. Incremental grid clustering is more efficient than usual clustering algorithms since they latter only update the previous feature space partition, instead of repartitioning the whole space whenever few points are added or removed. Then, the system merges these updated partitions in an effort to recognize the most dissimilar outliers. Incremental grid clustering usage contributes to lowering system complexity, which makes it more feasible for real-time detection.

Regarding SDNs and their challenges, like high density and variety of hosts, He et al. [102] recently developed a two-stage unsupervised clustering algorithm for anomaly detection. The first stage is a feature selection procedure used to remove unnecessary features in the dataset. Its basis is the calculation of a maximal information coefficient (MIC), which describes the relationship between two continuous features, and relevancy, which is a symmetric uncertainty estimator for discrete features. After selecting relevant features, a density peak-based clustering algorithm classifies the reduced dataset into normal and misbehaved patterns. Their experimental results proved that when a typical SDN hierarchy of controllers is used, the traffic data can be locally analyzed in each controller. This lessens the volume of traffic shuffled across the network.

Some of the main limitations of anomaly detection methods are basically: the absence of labeled data; finding of new unknown anomaly patterns; noisy data; and high false alarm rates. As an effort to overcome these problems, Bigdeli et al. [103] proposed an incremental two-layer cluster based structure for anomaly detection. The core idea is to cluster network data and represent these clusters as a Gaussian Mixture Model, so the model can categorize new instances and also detect and ignore redundant ones. Moreover, the high false alarm rate issue was addressed by a collective labeling method, which labels new inward instances in both collective and incremental ways.

5.2.1 Summary

With regard to clustering-based approaches, their advantages are listed as the following.

- Incremental clustering has a fast response generation.
- Stable performance when comparing to statistical methods or classifiers.
- Reduce computational complexity due to the ability to group large datasets into small ones.

However, limitations of these techniques can be seen below.

- They are highly dependent on proximity measures, and each one can affect the detection rate in a positive or negative way.
- Time consuming.
- They are not optimized for anomaly detection.
- Sometimes, the algorithms can be trapped in the local minima.

At last, Table 6 summarizes some characteristics of the discussed clustering approaches with regard to data precedence, investigated anomalies, and validation metrics used to test detection performance.

5.3 Finite state machine methods

A Finite State Machine (FSM), also called finite automata, is a mathematical behavioral model composed of states, transitions, and actions, used to represent computer problems or logical circuits. Each state stores information about the past, which are changes that have occurred since the entry into a state from the start of the system to the present time. This type of machine can only be in one state at a time. A transition indicates a state change and is disclosed by a condition, which must be achieved for the transition to occur. An action is a description of an activity, which must be carried out at a particular time. Moreover, these machines have strong analytical techniques, given that one can explore every possible sequence of states, since their alphabet of input and output allows representing a wide variety of situations.

Estevez-Tapiador et al. [104] presented a protocol anomaly detector using a finite state machine (FSM) approach, where network protocols were modeled from state sequences and transitions through a Markov chain. Its main idea was to monitor a given protocol in order to find deviations from "normal" usage. If the conditions are complete enough, the model can detect illegitimate behavioral patterns successfully.

Su [105] employed finite state machines to implement a framework applying frequent episode rules for a network intrusion prevention system (NIPS). The presented NIPS was developed to explore Probe attacks and anomalies that are difficult to be effectively detected by firewalls and anti-virus software. At first, it works by mining log files, which are posteriorly refined, resulting in episode rules that are converted to build an FSM. Via the FSM, every connection on a particular port is monitored and mapped out. Once a default alarm condition is achieved, the integrated real-time firewall update tool disconnects the malicious connection.

In [106], the authors produced an engineering method of gathering only a small volume of relevant IP flow records and aggregated them into a state space representation. This aggregation served as input to a finite state machine scheme. They developed an FSM with a stream learning component, such that it would be feasible to start modeling and learning a fine-grained communication profile in real-time. Their system produced promising detection rates over botnet malware detection. Additionally, they concluded that it is worthwhile to use limited IP flow data rather than large datasets for training.

5.3.1 Summary

Finite state machine techniques are not as popular as statistical or classification techniques, however, they have some good points to consider.

- Robustness and flexibility
- Strong analytical techniques, since their alphabet of input and output, allows representing a wide variety of situations
- High detection rate whether there is a considerable knowledge base regarding attacks and normal cases.

Some disadvantages of these techniques are listed below.

- Time-consuming.
- Inability to detect rare or indefinite attacks.
- Dynamic updating of rules/conditions are costly.

Table 7 summarizes some characteristics of the discussed clustering approaches, regarding data precedence, investigated anomalies, and validation metrics used to test detection performance.

5.4 Classification-based methods

Classification [107] is widely used in the anomaly detection field. The main idea of such techniques applied to this area can be summarized as two steps. First, during the training phase, a classifier is built (learned) using labeled training data. Then, this classifier is used to classify an instance as normal or anomalous (testing phase). According to each available labeled data for training, classification-based anomaly detection techniques can be either multi-class

Table 6 Comparison of clustering anomaly detection approaches

Paper	Year	Tech.[a]	Anomaly type[b]	Dataset[b]	Source[c]	Validation metrics
Rajasegarar et al. [97]	2014	k-NN clusters	Randomly generated set of anomalous data (S)	IBRL and GDI (R) and Banana and Gaussmix datasets (S)	–	ROC, detection rate (DR) and false positive rate (FPR)
Mazel et al. [98]	2011	Sub-Space clustering, Evidence Accumulation and Inter-Clustering	Few ICMP pkts, network scan (R)	Real traffic trace from the public MAWI repository of the WIDE project (R)	IP flow	Cluster similarity graph and outlier similarity graph for destination aggregated data
Karami and Guerrero-Zapata [99]	2015	K-Means, PSO	Abnormal source behavior, flooding attack (R)	UCI machine learning repository/CCNx data repository of Univ. of Politecnica Catalunya (R)	IP flow	Detection rate (DR–recall), FPR, precision, F-measure
Carvalho et al. [100]	2016	ACO (modified for clustering), DTW	DoS, DDoS, port scan, flash crowd (S)	University network (R) and simulated anomalies (S)	NetFlow	NMSE, accuracy, TPR, FPR, ROC curve
Dromard et al. [101]	2017	Incremental grid clustering algorithm (IGDCA)	RST and SYN attacks, and generic outliers (R)	ONTS dataset and MAWILab network traces (R)	IP flow	TPR, FPR
He et al. [102]	2017	Density peak based clustering algorithm	DoS, Probe, R2L, U2R (R)	KDDcup99 (R)	TCP dump	Classification accuracy
Bigdeli et al. [103]	2018	Spectral-based and density-based clustering	DoS, Probe, R2L, U2R (R/S)	KDDCUP99, Darpa98, NSLKDD, DataSetMe, and IUSTSip (R/S)	TCP dump, IP flow	ROC curve

[a]Clustering techniques/methods used
[b]Data precedence; R, Real; S, Simulated
[c]Source types in blank are either not clearly specified by the authors or not relevant in their research

Table 7 Comparison of finite state machine anomaly detection approaches

Paper	Year	Anomaly type[a]	Dataset[a]	Source[b]	Validation metrics
Estevez-Tapiador et al. [104]	2003	Protocol misusages (R)	TCP traffic filtered by destination port (R)	–	–
Su et al. [105]	2010	DoS, worm (R)	SMB with NetBIOS Session service (R)	–	–
Hammerschmidt et al. [106]	2016	Botnet malware (R)	Publicly available dataset of manually labeled IP flow traces (R)	–	TP, FP, Precision

[a]Data precedence; R, Real; S, Simulated
[b]Source types in blank are either not clearly specified by the authors or not relevant in their research

or one-class. The latter occurs when all training data have only one normal class label. The first assumes that training instances have multiple normal class labels. In this case, a classifier is built to be able to distinguish instances among normal classes and those who do not belong to any class (anomaly).

5.4.1 Naïve Bayesian

Naïve Bayesian is a simple probabilistic classifier commonly used for network intrusion detection problems. It combines prior information with sample information and implements it in statistical deduction, which uses probability to show all forms of uncertainty. Its principles are founded on the assumption that all input attributes are conditionally independent to each other. Thus, it calculates the probability of a certain instance belonging to a singular class.

Klassen and Ning [108] proposed a Naïve Bayesian approach to detect black holes, selective forwarding and DDoS attacks, in real time. The system monitored packets sent from nodes; therefore, their behavior is checked in order to detect any abnormality. The classifier assumes that data are normally distributed; then, the probability of a sample belonging to a class is calculated by a normal distribution probability procedure. Tao et al. [109] also used a Naïve Bayesian approach; however, they combined it with a time slicing function and, thus, they exploited the relationship between time and network traffic, since network traffic changes at distinct times and some traffic does not occur at a particular time. The work of Swarnkar and Hubballi [110] accurately detected suspicious payload content in network packets through the use of the one class Naive Bayes classifier for payload based anomaly detection (OCPAD), a combination of frequency information of short sequences with a one class multinomial naïve Bayes classifier.

5.4.2 Support vector machines

Another classification method is Support Vector Machine (SVM) [111], which is also used in pattern recognition. SVMs are a supervised learning concept characterized by the use of feature vectors/kernels (such as radial basis function—RBF), the nonexistence of local minima, sparseness of the solution, and capacity check achieved by operating on the border (the distance of the solution hyperplane to its closest point). Classifiers are obtained with good generalization, which is defined as its ability to correctly predict the class of new data from the same domain in which learning occurs.

Catania et al. [112] proposed a novel approach to providing autonomous labeling to normal traffic, in order to overcome imbalanced class distribution situations and reduce the presence of attacks in the traffic data used for training an SVM classifier. Amer et al. [113] applied two modifications of the unsupervised one-class SVM: Robust one-class SVMs and eta one-class SVMs. Their goal was to make the decision boundary less sensitive to outliers in the data.

Erfani et al. [114] stated that problem domains with a high number of dimensions are an obstacle to anomaly detection since irrelevant features can cover the presence of anomalies. Additionally, although the use of SVMs in detecting anomalies is effective on small datasets with many features, in complex high-dimensional data, the method is likely to take a long time for training. To overcome this limitation, the authors combined an unsupervised deep belief network (DBN) with one-class SVMs. The unsupervised DBN is trained to extract the features that are less sensitive to irrelevant deviations in the input data, producing a new data set suitable for being used to train a one-class SVM.

Additionally, Wang et al. [115] created an effective IDS based on a SVM with augmented features. Their framework integrates the SVM with the logarithm marginal density ratios transformation (LMDRT), a feature transduction technique that transforms the dataset into a new one. The new and concise dataset is used to train the SVM classifier, improving its detection. By evaluating the framework using the mostly used NSL-KDD dataset, the authors could achieve a fast training speed, high accuracy and detection rates, as well as low false alarm presences.

Kabir et al. [116] proposed an IDS based on a modification of the standard SVM classifier, known as the least square sup-

port vector machine (LS-SVM). This alteration is sensitive to outliers and noise in the training dataset when compared to a regular SVM. Their decision-making process is divided into two stages. The first stage is responsible for reducing the dataset dimension by selecting samples depending on the variability of data by using an optimum allocation scheme. Then, the next stage uses these representative samples as the input of the LS-SVM. The algorithm was optimized to work on both static and incremental data and produced effective results.

5.4.3 Artificial neural networks

Artificial Neural Networks (ANNs) are computational techniques that present a mathematical model inspired by the neural structure of intelligent organisms, which acquire knowledge through experience. They are self-adapting, self-organizing and able to learn according to inputs and feedback from the ecosystem within which they operate. Although neural networks are considered a bio-inspired model, they are used in the anomaly detection domain mostly as classifiers. Multi-layer Perceptron (MLP) and Back Propagation (BP) algorithms are the most common ANN techniques.

Subba et al. [117] employed an ANN model in order to introduce an intelligent agent for classifying whether the underlying patterns of audit records are normal or abnormal while classifying them into new and unseen records. This goal is accomplished through feed forward and back propagation (BP) algorithms. They are responsible for feeding the neural network with inputs processed to become vectors, comparing the calculated and expected output generated by the ANN, and at finally, altering the weights of the ANN connections in order to approximate the output. After some experiments, this approach proved to be high in performance and low in terms of computational overhead.

Saeed et al. [118] proposed a two-level anomaly-based IDS using a Random Neural Network (RNN) model in an IoT environment. The RNN model was employed in order to build a behavior profile based on both valid and invalid system input parameters to distinguish normal and abnormal patterns. The system learns the relationship between input and output by adjusting the interconnection weights of the RNN. The second level of the IDS is responsible for detecting a broad range of Illegal Memory Access (IMA) bugs and data integrity attacks.

Brown et al. [119] proposed a two-class classifier using an evolutionary general regression neural network (E-GRNN) for intrusion detection based on the features of application layer protocols such as HTTP, FTP, and SMTP. Authors used evolutionary computation to evolve parameters and salient features (feature mask) from the general regression neural network and to find its optimal configuration. This method reduces computational complexity by eliminating unnecessary features and increases classification accuracy.

Supervised learning models can train a classifier by only using labeled samples, which are difficult to obtain due to requiring expert knowledge. On the other hand, unsupervised approaches consider only unlabeled samples, which are easily available in real-world situations. Ashfaq et al. [120] proposed a fuzziness-based semi-supervised learning approach, merging both unlabeled and labeled data to build a better classifier. The base classifier was the neural network with random weights (NNR_W) due to its excellent learning feature. For all unlabeled samples produced by the NNR_W, their model computes fuzziness as an effort to discover relationships between the output fuzzy membership vectors and misclassification rates. Subsequently, the unlabeled samples receive a predicted label according to fuzziness groups (high, mid and low), and the classifier is retrained with them. The authors found out that samples within low and high fuzziness groups are vital in improving the performance of the NNR_W classifier and result in high accuracy rates. Additionally, samples belonging to mid-fuzziness groups showed increased uncertainty of misclassification.

5.4.4 Ensemble approach

An ensemble approach means combining responses of multiple classifiers into a single one, thus yielding better performance compared to using individual classifiers. The weighted-majority algorithm (WMA) is a well-known ensemble technique responsible for combining and selecting the best response among all classifiers [121].

Aburomman and Reaz [122] cite that an ensemble classifier achieves success conditional to the diversity in the outcomes of its component classifiers and the method chosen to combine these outcomes into a single one. In this manner, they first trained six SVM experts (in which an expert consists of five binary classifiers producing a binary vector of outcomes) and six other experts using the k-nearest neighbor (k-NN). Then, they used particle swarm optimization (PSO), meta-optimized PSO and weighted majority algorithm (WMA) techniques to combine the experts' opinions and accurately create three new ensembles. After testing and comparing the three new techniques over some KDD99 datasets, the PSO ensemble approach achieved better results, improving accuracy by 0.756 %, in a short runtime. The authors explained that the sets of generated weights, which were also optimized to produce results with the best possible accuracy, were responsible for the success of the PSO-based ensemble. Despite the fact that the meta-optimized PSO approach accomplished a better accuracy gain, it took 500 times more time to achieve it. On the other hand, the WMA approach had the worst results since it had a reasonably low

accuracy of base classifiers for the occurrences of Normal and R2L classes.

Sornsuwit and Jaiyen [123] proposed a novel ensemble approach for intrusion detection using the AdaBoost algorithm, which combines the solution of the following classifiers: naïve Bayes, decision tree, multilayer perceptron (MLP), k-NN and SVM. The AdaBoost algorithm initializes the distribution of data, trains the classifiers, evaluates errors and assigns weights to each of them. Then, the combination of classifiers is linear and based on a weighted voting approach.

Bukhtoyarov and Zhuckov [124] developed an ensemble-distributed classifier for network IDS based on a new tree-level approach for combining the individual classifiers' decisions. The approach relies on using ensembles of neural networks designed through genetic programming-based ensembling (GPEN). GPEN automatically builds a program using genetic programming operators to indicate how to combine the component networks' predictions in order to get a reliable ensemble prediction. This study differs from others dealing with traditional ensemble since it provides the partial obtaining of adaptive outcomes by distinct classifiers deprived of an ensemble classifier.

5.4.5 Summary

To sum up, classification-based methods are prevalent due to its simplicity and effectiveness. Here are some additional advantages.

- Flexibility for testing and training by incorporating new information into the execution strategies.
- High detection rates for acknowledged attacks.
- Artificial Neural Networks have an adaptive nature, being possible to train and test cases incrementally.
- Regarding efficiency, multi-level neural network techniques are better than a single-level neural network.
- Ensemble methods perform well by combining multiple classifiers, even if they are weak ones.

However, despite being popular among researchers, there are some disadvantages, as follows.

- High resource consumption.
- Inability to detect unknown anomalies without some relevant training information.
- Neural network usage may cause over-fitting.
- The selection of sample datasets is slow for big datasets.
- In some cases, real-time performance is hard to acquire.

Table 8 summarizes some characteristics of discussed clustering approaches, regarding data precedence, investigated anomalies, and validation metrics used to test detection performance.

5.5 Information theory

Information Theory is a mathematical subject centered on the quantification of information and redundancy analysis. It was formerly envisioned by Claude E. Shannon, in 1948, while seeking data compression, transmission, and storage for signal processing and communication operations [125]. However, its application extended to many other purposes such as telecommunications, estimation, decision support systems, pattern recognition and so on [126]. There are several information-theoretic measures, such as Shannon entropy, generalized entropy, conditional entropy, relative entropy, information gain and information cost.

Its use for anomaly detection purposes relies mainly on the calculus of mutual information or entropy values for designated traffic features in order to identify anomalous distributions on them. Since it adopts statistical properties for the time series of a traffic-related features (e.g. Gaussian), this methodology may result in inaccuracies.

5.5.1 Entropy

Entropy is the most well-known information theoretical measure, defined as the equivalent probabilities, or the uncertainty, involved in the value of a stochastic variable or the occurrence of a random process. Considering the use of entropy in the anomaly detection field, it is efficient in describing traffic features, such as source/destination ports or IP addresses, as distributions, since there are certain types of anomalies causing significant disturbances on these distributions. In this manner, it is possible to detect, for instance, a port scan attack, indicated by a change in the entropy of destination ports, or even the occurrence of a DDoS attack, denoted by changes in the entropy of source/destination IP addresses [127].

David et al. [128] proposed an enhanced detection of DDoS attacks through a fast entropy method and the use of flow-based analysis. Authors aggregate the observed flows into a single one with consideration to the flow count of each connection at a certain time interval instead of taking the packet count of every connection. The second step is basically the calculation of the fast entropy of the flow count for each connection. Finally, an adaptive threshold is generated based on the fast entropy and the mean and standard deviations of flow counts. The constant update of the threshold with regard to the traffic pattern condition improves detection accuracy, while fast entropy use reduces computational processing time.

Amaral et al. [129] proposed a feature-based anomaly detection system using both IP Flow properties and a graph

Table 8 Comparison of classification-based anomaly detection approaches

Paper	Year	Tech.[a]	Anomaly type[b]	Dataset[b]	Source[c]	Validation metrics
Swarnkar and Hubballi [110]	2016	Naïve Bayesian	Buffer overflow, shell-code attacks	Network of IIT Indore (R) and HTTP attack dataset (R)	–	Detection rate, FPR
Klassen and Ning [108]	2012	Naïve Bayesian	Black Holes, selective forwarding, DDoS (S)	NS2 simulated network traffic data (S)	–	Confusion Matrix (TP FP precision recall F-measure)
Tao et al. [109]	2008	Naïve Bayesian	Scan attack, DoS, ARP attack, Fragment attack, and comprehensive attack (R)	DARPA1999 (S)	TCP dump	Average detection rate
Kabir et al. [116]	2017	LS-SVM	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	Precision, recall, F-value, probability of detection, probability of correct detection, FPR, accuracy
Wang et al. [115]	2017	SVM	DoS, Probe, R2L, U2R (R)	NSL-KDD (R)	TCP dump	Accuracy, DR, false alarm rate
Erfani et al. [114]	2016	SVM	Outiliers (R)	UCI machine learning repository (R) and two synthetic datasets (S)	–	ROC and area under the curve (AUC)
Catania et al. [112]	2012	SVM	Generic attack distributions (R)	1998 DARPA (S)	TCP dump	Attack detection rate (DR) and false alarm rate (FA)
Amer et al. [113]	2013	SVM	Outiliers (R)	UCI machine learning repository (R)	–	Area under the ROC curve (AUC) and ROC curves
Subba et al. [117]	2016	Artificial Neural Networks	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	Accuracy, DR
Saeed et al. [118]	2016	Random Neural Networks	Data integrity attacks and ilegal memory access (S)	Wireless sensor nodes–based IoT system (S)	–	Accuracy, FPR, FNR, TPR, TNR
Brown et al. [119]	2016	General Regression Neural Network	Generic anomalous instances (S)	UNB ISCX dataset (S)	Features of application layer protocols	Accuracy, DR, FNR, TNR, FPR
Ashfaq et al. [120]	2017	Neural Network with random weights	DoS, Probe, R2L, U2R (R)	NSL-KDD (R)	TCP dump	Accuracy
Aburomman and Reaz [122]	2016	Ensemble (SVM, k-NN, PSO, WMA)	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	Accuracy
Sornsuwit and Jaiyen [123]	2015	Ensemble (Adaboost, Naïve Bayes, MLP, SVM, DT)	R2L, U2R (R)	KDD99 (R)	TCP dump	Sensitivity, Specificity
Bukhtoyarov and Zhuckov [124]	2014	Ensemble (GPEN, neural networks)	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	DR, FPR

[a]Classification techniques/methods used
[b]Data precedence; R, Real; S, Simulated
[c]Source types in blank are either not clearly specified by the authors or not relevant in their research

representation in order to carry out a deep inspection of network traffic. The detection is based on the Tasallis entropy, a generalization of Shannon entropy. The major divergence is that it has a parameter to define which probabilities will contribute to the entropy result. It adjusts the sensibility of the anomaly detector, allowing it to adapt to different types of networks and detect more inexpressive attacks than those detected by methods based on volume analysis.

The work presented by Bhuyan et al. [130] brings an outlier-based anomaly detection approach using generalized entropy and mutual information for creating a feature selection technique capable of choosing a relevant, non-redundant subset of features. According to the authors, since mutual information reduces the uncertainty about one random variable and generalized entropy measures the amount of uncertainty in the data, they make detection faster and more accurate.

Moreover, Berezinski et al. [131] introduced a network anomaly detector, based on Shannon entropy, in order to detect modern botnet malware. Their approach created a network profile, which stores min and max entropy values in a sliding time window of 5 minutes. These values were used for comparison with the observed entropy. This defines a threshold, thus, abnormal dispersion or concentration for different feature distributions can be identified. Finally, the authors used popular classifiers, such as decision trees and Bayesian networks, in order to classify the anomalies.

Behal and Kumar [132] stated that since DDoS attacks and flash events cause substantial alterations in network traffic patterns, information theory-based entropy or divergence can rapidly capture such disparities in network traffic behavior. Therefore, they proposed a generalized anomaly detection algorithm, which exploits the entropy difference between traffic flows. They employed a set of generalized ϕ-Entropy and ϕ-Divergence metrics, in which the detection efficiency was directly connected to the information distance between legitimate and attack traffic. The proposed algorithm resulted in high detection accuracy with regard to flash events and High-Rate DDoS, overcoming the results of other information theory approaches in the literature.

5.5.2 Kullback–Leibler distance

The Kullback–Leibler Distance or Divergence (KLD) measures the difference between the true probability distribution P and an arbitrary probability distribution Q (an approximation of P).

The work of Xie et al. [133] consisted of an algorithm to track long-term anomalies in WSNs by using the Kullback–Leibler divergence to measure the differences between global Probability Density Functions (PDF) for each of two consecutive periods of time. This function produces a time series to be analyzed and make decisions based on the adaptive threshold, identifying any unusual changes. The approximate Kullback–Leibler divergences, obtained from distributed computing with no significant accuracy degradation, is used to reduce the communication cost since it can reflect the variation among PDFs in a sensitive manner. Li and Wang [134] proposed a differential Kullback–Leibler divergence based anomaly detection scheme for wireless sensor networks. The authors used a clustering approach to separate the sensor nodes into clusters. All the nodes composing a cluster had related sensed value and were physically close to each other. Then, the Kullback–Leibler divergence was used within each cluster in order to detect abnormal values by statistically measuring the disparity between two data sets. Their work achieved a good detection rate and low false alarm rate while consuming less energy than other similar studies in the literature.

5.5.3 Summary

In conclusion, information theoretic-based approaches have been emerging increasingly in the network anomaly detection field. Their main benefits are that they can be highly scalable, very sensitive and low to false positives. Other advantages are stated below.

- Operating in an unsupervised mode is possible.
- There are no assumptions about the primary statistical distribution for the data.
- Since information theory-based methods only use header information for calculation, the complexity of time and space is a minor problem.

Besides, they are susceptible to these limitations.

- The adoption of statistical properties for the time series of traffic-related features (e.g., Gaussian) may cause inaccuracies.
- The detection of anomalies may be possible only if there is a significant presence of them in the data set. This way, these approaches need a highly sensitive information theoretic measure to detect irregularities made by very few anomalous patterns.
- Difficulty in associating an anomaly score with a trial case.

Table 9 summarizes some characteristics of discussed clustering approaches, regarding data precedence, investigated anomalies, and validation metrics used to test detection performance.

Table 9 Comparison of information theory anomaly detection approaches

Paper	Year	Tech.[a]	Anomaly type[b]	Dataset[b]	Source[c]	Validation metrics
David et al. [128]	2015	Entropy	DDoS (R)	CAIDA dataset (R)	IP flow	Empiric entropy variation analysis
Amaral et al. [129]	2017	Entropy	DDoS, alpha flow, portscan, network scan (R/S)	Universities traffic (R)	IP Flow	TPR, FPR
Bhuyan et al. [130]	2016	Entropy	DoS, probe, R2L, U2R (S)	Testbed dataset (S), KDDcup99, NSL-KDD, UCI ML repository datasets (R)	Packet/flow records	Detection rate, precision, recall, f-measure
Berezinski et al. [131]	2015	Entropy	Port scan, DDoS (S)	Legitimate traffic from medium size network (R)	IP flow	Correlation, Accuracy, FPR, TPR, ROC
Behal and Kumar [132]	2017	Entropy	DDoS, flash events (S)	MIT Lincoln dataset, FIFA, DDoSTB, and CAIDA datasets (R) and D-ITG traffic generator, Bonesi (S)	IP flow	DR, precision, FPR, TNR, NPV, F-measure, F-measure complement, Classification rate
Xie et al. [133]	2017	Kullback–Leibler	Long-term anomalies	RSS measurement from UMICH network (R)	–	FPR, Accuracy, ROC
Li and Wang [134]	2012	Kullback–Leibler	Generic anomaly data values (R)	Real sensed data (R)	–	Detection rate, FPR

[a]Information theory techniques/methods used
[b]Data precedence; R, Real; S, Simulated
[c]Source types in blank are either not clearly specified by the authors or not relevant in their research

5.6 Evolutionary computation

The field of evolutionary computation, also named bio-inspired computing [135], is a set of intelligent algorithms and methods inspired by natural evolution and able to learn and adapt like biological organisms [136]. It encompasses genetic algorithms (GA), genetic programming (GP), evolution strategies (ES), particle swarm optimization (PSO), and artificial immune systems (AIS) [137,138].

5.6.1 Artificial Immune Systems (AIS)

Artificial Immune Systems (AIS) are adaptive systems, enhanced by theoretical immunology and biological immune system functions, principles and models, and are applied to problem-solving, as defined by de Castro and Timmis [139].

The authors of [140] presented EPAADPS, a proactive anomaly detection and prevention system based on an Artificial Immune System (AIS) aiming to identify and prevent new and undetected anomalies. Their motivation lied on gaps found in related previous studies, such as the lack of co-operation between detectors in order to classify any pattern as anomalous, the identification and inhibition of novel and zero-day attacks and lacks in self-configuration, learning, adaptability and preventive abilities. The whole system consists of three modules: the repertoire training module (RTM), responsible for selecting efficient detectors to generate a detector set (DS); the vulnerability assessment module (VAM), which creates collaborative detector agents (DA) able to correctly identify and flag any test set instance happening to be an anomaly; and response module (RM), which takes appropriate preventive actions in the cases where VAM found an anomalous instance. Saurabh and Verma also combined PCA and min-max normalization as a pre-processing feature in order to make the dataset both substantial and stable, respectively. In this manner, the number of features is reduced, helping the choice of better-trained detectors.

Moreover, Igbe et al. [141] proposed a distributed NIDS against cyber-attacks using the Negative Selection Algorithm (NSA), which exist in the AIS field. The entire system has autonomous agents communicating with each order while running the NSA to create classification rules. These rules and identified threat vectors are shared among all agents and enhance the fast detection of more problems.

Shahaboddin et al. [142] introduced Co-FAIS, a cooperative-based fuzzy artificial immune system for detecting malicious activities in a WSN. The adopted defense strategy is modular and derived from the danger theory of the human immune system as an AIS. The agents work in a mutual way in order to identify attackers or any abnormalities in sensor behavior regarding the context antigen value (CAV). Then, agents inform the Fuzzy Q-learning algorithm initiation threshold, which examines the attack behavior and checks if the system can respond and defend itself. That response was designed to act similar to the ability of rapid response to recurring attacks present in a natural immune system. The response module elaborates an attack signature and eliminates it from the safe list; therefore, if repeated, the reaction to the same attack will be quicker.

5.6.2 Genetic algorithms (GA)

Genetic algorithms (GA) are commonly used as part of a whole intrusion detection system together with other techniques. As in [143], the authors use a genetic algorithm to transform the data set such that an SVM classifier can better process it. In [131], for instance, the authors combine a genetic algorithm (GA) with kernel principal component analysis (KPCA). The genetic algorithm creates a new optimal set of features and assigns a separate group with a certain priority to each obtained feature.

In their research [144], Singh and Kushwah employed genetic algorithms to build an optimized cluster-based intrusion detection system in wireless sensor networks. The entire system was divided into four modules: the data collection module, which makes the head node observe the movement of the member sensor node; the intrusion information module, which gathers intrusion information for explanation; the intrusion detection module, responsible for setting a device activity as the misbehavior or legitimate behavior prior to a threshold; and the alert module, in which the cluster-head node alerts nearby nodes about the existence of intrusion. A genetic algorithm is used to mutate the nodes presenting less energy by a mutation parameter with a mutation probability in order to flip the node energies; thus, power consumption and network efficiency are improved.

Another approach using Genetic Algorithms is presented by Hamamoto et al. [145]. The GA is used to deal with uncertainties in network traffic, and through natural selection, learn the normal characteristics of the traffic flows. As all traffic attributes used in the research are numeric, the authors applied a numeric chromosome encoding to optimize each time interval separately and each attribute in parallel. The result is the Digital Signature of Network Segment using Flow Analysis (DSNSF), a prediction of the network traffic behavior in each time interval. Moreover, the authors also added a Fuzzy logic approach to assess whether a time interval in the IP flows data has an anomaly or not. The evaluation was conducted by using a real network traffic from an university with simulated anomalies injected in some flow entries.

5.6.3 Differential evolution

Differential evolution is a global search evolutionary algorithm also used for detecting anomalies. Although it is not widely used yet, it has great potential in order of being wor-

thy to mention in this section. It encompasses two concepts within the area: the idea of using larger population from genetic algorithms and self-adapting mutation from evolutionary strategies. Elsayed et al. [146] applied a feature reduction mechanism using a flexible neural tree to select significant traffic features and then adopted a differential evolution algorithm to evolve individual (rules) for anomaly detection. A fitness function calculates the quality of every rule or individual.

5.6.4 Particle swarm optimization

Particle Swarm Optimization is a common evolutionary computation technique used for anomaly detection. Its main purpose is to perform an optimum search, and that is why this algorithm is mainly combined with clustering techniques and classifiers, such as k-means [38,99] and SVM [122,147,148], for instance (please refer to works discussed in sections 6.2 and 6.4).

Bamakan et al. [149] proposed a novel intrusion detection framework by using a modification of the PSO, called time-varying chaos particle swarm optimization (TVCPSO). It is a new adaptive, robust, precise optimization method, aimed at doing parameter setting and feature selection for multiple criteria linear programming (MCLP) and SVM simultaneously. The authors introduced time varying inertia weight and a time varying acceleration coefficient, along with the adoption of the chaotic concept in the PSO. In this manner, the PSO algorithm searches the optimum faster than normal, while avoiding the search being stuck to a local optimum.

5.6.5 Summary

Evolutionary computation methods are increasingly obtaining distinction due to their intelligent algorithms that can learn and adapt like real living organisms, therefore being able to produce latent solutions to many of the complex network problems that have been intensified recently. Other advantages of these methods are the following.

- They add to intrusion detection systems capabilities for parallel processing.
- Prior knowledge of the problem space is not required.
- The natural retraining ability makes the entire system more adaptable.
- Noise and discontinuities existing in the dataset do not cause a considerable impact on solutions.

Although their efficiency, evolutionary methods also have some limitations.

- The fitness function may not be trivial to find.
- Choosing the optimal parameters is hard.
- Sometimes, it can be a complicated task to map the problem into a biological approach.

Table 10 summarizes some characteristics of the discussed evolutionary computation approaches, with regard to data precedence, aimed network paradigm, techniques, anomalies, and validation metrics used to test detection performance.

5.7 Hybrid/others

This section presents hybrid approaches to anomaly detection, which are a combination of various classes of algorithms, techniques, and methods. Additionally, unclassified techniques, which are not listed in previous sections but are still interesting and promising, are also listed here.

Grill and Pevný [151] state that successive alarm analysis is costly and cannot cover all alarms, only a small portion, as well as the noise in training data is always an important feature to consider. Moreover, combining anomaly detectors, although simple, may become a significant challenge when it attempts to combine the output of individual detectors. Therefore, the authors propose a novel approach to finding a convex combination of various anomaly detector outputs and carried out a study on the effects of label noise in the training dataset over the accuracy of combinations achieved by different detectors. They compare their approach to two existing ensemble methods, one using NetFlow and the other using HTTP server logs.

Another interesting hybrid intrusion detection system is proposed by Al-Yaseen et al. [152], in which the authors combine the SVM and extreme machine learning (EML) classifiers and the k-means clustering technique. The classifiers are responsible for reducing false positives as well as improving detection accuracy. The categories of attacks are divided into three groups; four SVMs classify instances as DoS, U2R, R2L, or Normal while an ELM classifier detects probe attacks, since they are better for them than an SVM. On the other hand, k-means is modified to build a suitable training dataset, which can meaningfully contribute to improving the classifiers' training time and overall performance. The modification consists of selecting the initial centroids of clusters conditional to the maximum distance between them and dataset instances. Five datasets are produced to each one of the five classification categories and serve as the basis of creating accurate SVM and ELM classifiers.

Forestiero [153] used a swarm intelligence technique to build a bio-inspired clustering algorithm in order to identify anomalies in distributed data streams. Bio-inspired agents follow the principles of the flocking-based examination approach, which states that agents will interact autonomously with immediate neighbors and form flocks (clusters) of similar agents. The similarity between agents depends on the

Table 10 Comparison of evolutionary computation anomaly detection approaches

Paper	Year	Tech[a]	Anomaly type[b]	Dataset[b]	Source[c]	Validation metrics
Saurabh and Verma [140]	2016	AIS	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	FPR, DR
Igbe et al. [141]	2016	AIS	DoS, Probe, R2L, U2R (R)	KDDTrain+20% (R)	TCP dump	DR, FPR
Shahaboddin et al. [142]	2014	AIS, FQL	DDoS (S)	NS-2 simulation (S)	UDP traffic	Accuracy, FNR, FPR
Singh and Kushwah [144]	2016	GA	WSN node issues	Sensor Field of Area 100×100 m (R)	–	Packet delivery rate, end-to-end delay, distance vector, system lifetime and throughput
Hamamoto et al. [150]	2017	GA	DoS, DDoS, Flash crowd (S)	University dataset (R)	NetFlow	Accuracy, precision, recall, F-measure, FPR, ROC AUC, Mis. Rate
Elsayed et al. [146]	2015	DE	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	DR, FPR, FNR
Bamakan et al. [149]	2016	PSO	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	Accuracy, DR, FPR

[a]Evolutionary computation techniques/methods used
[b]Data precedence; R, Real; S, Simulated
[c]Source types in blank are either not clearly specified by the authors or not relevant in their research

carried data items and can be calculated using various techniques such as measuring the Euclidian distance of associated data items.

Salem et al. [154] developed a framework for anomaly detection that operates in wireless body area networks (WBAN). They combined the SVM classification algorithm with the statistical linear regressive model. The SVM part classifies incoming sensor data as normal or abnormal. Then, whenever an abnormality is found, a linear regressive prediction model analyzes it and decides whether the patient is entering a dangerous state or a sensor is reporting incoherent readings. This decision is accomplished by building a decision tree and searching for linear coefficients from normal vital signs falling inside a given threshold.

Wang et al. [155] combined three classes of algorithms for the purpose of introducing a data abstraction phase situated between the well-known attribute construction and detection of model building phases that most IDSs have. Their idea was applied to process big data by reducing the amount of data while keeping the valuable information they carry. For that purpose, three strategies were proposed and evaluated. The attribute abstraction strategy was based on applying PCA for reducing the data to a low dimensional subspace and then projecting the testing data onto it in order to detect anomalies. The attribute selection strategy consists of calculating the information gain (IG) to rank the correlations of each attribute to the class—whether it is normal or attack—and select key attributes based on this ranking. After the selection, the authors combined this with a k-nearest neighbor and a PCA or SVM-based detection approach. Finally, the exemplary extraction strategy uses either k-means or affinity propagation clustering techniques to extract exemplars from the large audit data. After the extraction, authors also combined this with a k-nearest neighbor, a PCA or an SVM based detection approach.

Adaniya et al. [156] created a hybrid anomaly-based clustering approach for anomaly detection, combining the k-harmonic means (KHM) clustering method with the bio-inspired heuristic firefly algorithm (FA). The traffic profile is created through the GBA tool by using the historical traffic data, proposed by Proença et al. [157]. KHM solves the initialization sensitivity of k-means and the FA helps it converge to local optima. This approach groups data points in order to separate normal from abnormal ones. They achieved good outcomes, with true-positive rates above 80 % and false-positive rates below 20 %.

Chen et al. [158] managed to build a novel classifier through an evolutionary computation basis for intrusion detection. The central segment of this IDS is an artificial immune system (AIS), which is enhanced by a population-based incremental learning (PBIL) procedure. The PBIL enhancement in the AIS consists of evolving new antibodies with higher affinities than older ones, which are not capable of

properly recognizing the class (removal of weak antibodies). Then, the authors combined the AIS-PBIL with collaborative filtering (CF) in order to cluster all antibodies related to a target-occurrence and categorize target intrusions.

Bostani and Sheikhan [61] proposed a novel hybrid IDS framework consisting of anomaly-based and specification-based modules. Their goal was the detection of two routing attacks that cause significant problems in IoT: sinkhole and selective-forwarding attacks. The framework is divided into three stages. In the first stage, the specification-based agents in the router nodes identify suspicious nodes by analyzing the behavior of their host nodes and sending it to the root node. In the second stage, an anomaly-based agent located in the root node uses that information to extract traffic features and create samples for each source node. This is accomplished by using an unsupervised optimum-path forest algorithm and a MapReduce architecture for projecting clustering models. Finally, the last stage uses the first stage results to make decisions about mistrustful behavior detected in the second stage through a voting mechanism.

Grill et al. [159] propose a local adaptive multivariate smoothing (LAMS) method to effectively smooth an anomaly detector output in order to reduce the rate of unstructured false positives by the Nadaraya–Watson estimator. It replaces the output of a networking event with an aggregate of its output on similar network events observed previously.

Guo et al. [160] combine both misuse detection and anomaly detection to build an IDS. The development is divided into 2 phases. The first phase is the elaboration of a lightweight misuse detector based on the change of location of cluster centers. In phase 2, two anomaly detectors are built using the k-nearest neighbor (k-NN) algorithm. By this combination, authors were capable of detecting both known and unknown anomalies with a low false positive rate (FPR).

As an emergent network paradigm, Software-Defined Networks (SDN) also face the problem of DoS attacks, since massive malicious requests can truly harm their centralized control characteristic. Thus, although many researchers propose detection mechanisms, most of them only focus on detection itself. In this manner, Assis *et al.* [150] proposed GT-HWDS, a hybrid autonomic defensive approach for SDNs against DoS/DDoS attacks by applying a game theory (GT) decision-making model together with their Holt–Winters-based anomaly detection system (HWDS [67]). The GT-HWDS system is fully able to detect, identify and mitigate events of DoS/DDoS in SDN traffic. Their core contribution is the mitigation module performed by a GT-based method. GT consists of changing a problem with opposing interests into a game, where many "players" take actions to optimize the results of trying to achieve their objectives. Therefore, the system analyzes a set of probable actions for both attacker (malicious nodes) and defense systems, estimates rewards and costs for all measures, and finally, performs an optimal countermeasure. This blocks (mitigates) any traffic originating from the attackers' IP and port.

5.7.1 Summary

In summary, hybrid techniques are an excellent choice for situations in which the same system might solve many distinct problems. Also, one technique may overcome the limitations of others, leading to a more reliable system. These are the main advantages of hybrid anomaly detection methods.

- Hybrid methods can benefit from the main features of both anomaly and signature-based approaches.
- They can detect both known and unknown anomalies.

However, when developing hybrid methods, there some limitations to consider.

- As more techniques are used together, the is a high demand for computational resources, increasing its cost.
- Dynamism is still an unsolved problem.

Table 11 summarizes some characteristics of the approaches discussed in this section, with regard to data precedence, aimed network paradigm, techniques, investigated anomalies, and validation metrics used to test detection performance.

6 Open issues

There are a significant number of challenges within the anomaly detection field. This section aims to summarize the most relevant open issues found during the development of this paper and also to consider those most discussed in the literature. All of them were identified by analyzing and comparing all surveys [9,12–18] listed in Table 1 and every research addressed in this survey. The list and a brief discussion upon each topic can be seen below:

- *The concept of normality* It is one of the main steps to build a solution to detect network anomalies. The question "how to create a precise idea of normality?" is what has driven most researchers into creating different solutions through the years. This can be considered as the main challenge related to anomaly detection and has not been entirely solved yet. Many of the works discussed in this survey tried to achieve this goal.
- *Adaptability* Anomalies keep changing every time new ones are introduced or old ones are improved to overcome current detection solutions. Therefore, IDSs need to be constantly updated in order to adapt to those changes, and this is not an easy task.

Table 11 Comparison of hybrid/unclassified anomaly detection approaches

Paper	Year	Tech[a]	Anomaly type[b]	Dataset[b]	Source[c]	Validation metrics
Al-Yaseen et al. [152]	2017	SVM, EML, k-Means	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	FPR, detection rate, accuracy
Bostani and Sheikhan [61]	2017	Optimum-path forest algorithm	Sinkhole attack, selective-forwarding attack	WSN simulator (S)	–	Accuracy, TPR, FPR
Grill and Pevný [151]	2016	Classification methods	Portscans, SSH brute force (R)/ZeroAccess and other malwares (R)	Czech Technical University network (R)/30 arbitrary companies traffic data (R)	NetFlow/HTTP logs	Precision and recall
Forestiero [153]	2016	Flocking algorithm	Generic anomalies	Gauss and STREAM (S); UCI Machine Learning Repository and 1998DARPA (S)	TCP dump	Normalized mutual information (NMI), Precision and recall
Salem et al. [154]	2014	SVM and Linear Regression	Outlier detection (R)	Physionet database (R)	–	FPR, TPR, ROC
Wang et al. [155]	2016	PCA, IG, AP, k-means, SVM, k-NN	http common attacks, DoS, Probe, R2L, U2R (R)	Real http traffic dataset and KDD99 (R)	–	Detection rate, and execution time
Adaniya et al. [156]	2013	K-harmonic means, firefly algorithm	DoS, Flash Crowds (R)	University network (R) and simulated anomalies (S)	SNMP	
Chen et al. [158]	2016	AIS, population-based incremental learning, collaborative filtering	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	Accuracy,
Grill et al. [159]	2017	Local adaptive multivariate smoothing (LAMS)	Generic artificial attacks (S) and malwares (R)	Czech Technical University (CTU) network (R) and varyed HTTP database (R)	NetFlow, HTTP proxy logs	AUC,
Guo et al. [160]	2016	Location of cluster centers, k-NN	DoS, Probe, R2L, U2R (R)	KDD99 (R)	TCP dump	DR, TNR, FPR, Accuracy
Assis et al. [150]	2017	Game Theory (GT), Holt–Winters	DoS, DDoS (S)	University network traffic (R)	NetFlow	Precision, accuracy, drop rate

[a]Techniques/methods used
[b]Data precedence; R, Real; S, Simulated
[c]Source types in blank are either not clearly specified by the authors or not relevant in their research

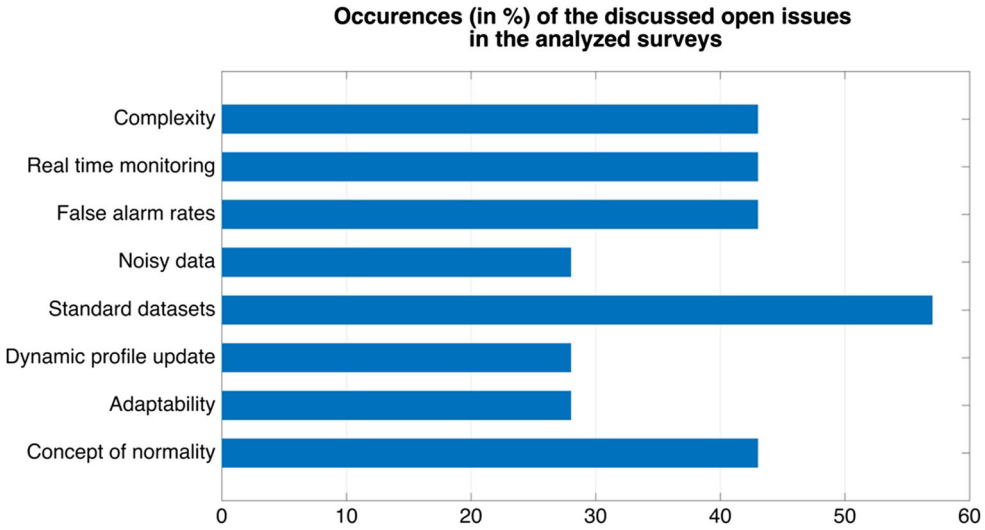


Fig. 10 Occurrences (%) of discussed open issues in the analyzed surveys

- *Dynamic profile update* Whenever an unknown attack is detected and addressed by anomaly-based IDSs, the profile database needs to be updated with these new data. Nevertheless, it is a challenge to carry out such updates dynamically, without compromising performance and generating conflicts.
- *Standard datasets* There are only a few openly available intrusion datasets with enough information about attacks; however, none of them is a standard evaluation dataset for anomaly detection. The lack of reliable public standard datasets, which can simulate accurate network environments, is still a problem.
- *Noisy data* Normal variations in datasets are also a problem when creating a profile since they can be misunderstood as abnormalities if they are not well defined. Moreover, this information is neither always clear in public datasets nor private ones.
- *False alarm rates* Another problem is to keep false alarms as minimal as possible; although it is still not possible to completely avoid them and build a one hundred percent reliable IDS. That still remains a challenge.
- *Real-time monitoring* The amount of traffic generated by computer networks today is constantly increasing as Internet traffic doubles every year. Therefore, it has been difficult to produce a reliable monitoring process on a network, in real time.
- *Complexity* As researchers try to cover all the challenges mentioned above, the complex nature of the systems increases by adding and mixing different techniques and approaches. Additionally, regarding data collection and preprocessing, the complexity of today's network architectures also contributes to the persistence of this issue.

The graph in Fig. 10 shows the relevance of each open issue discussed in this section amongst the others analyzed surveys [9,12–18] presented in Table 1, showing the most concerning issue in the anomaly detection field based on the study in this survey. For instance, the complexity issue appeared in 43% of the other surveys, while the standard dataset problem was considered in 60%. So, it can be observed that although many of the other discussed topics had a significant rate of discussion, the problem of not having a standard and updated dataset that simulates a real environment and contains labels for anomalies is a major concern among practitioners in the literature.

7 Conclusion

This literature review aimed to provide a theoretical understanding of the anomaly detection problem with regard to the different aspects related to it. It also aimed to provide a comparative analysis of different techniques developed to address this problem.

A discussion on what an anomaly is and the identification of its most common manifestations were presented. Based on their nature, anomalies can be grouped into point anomalies, collective anomalies, and contextual anomalies. Nevertheless, based on their causal aspect, they can also be divided into operational events, flash crowd, measurement anomalies, and network attacks. Their correct identification is crucial to the development of an IDS, which can focus on the primary constraints related to each kind of occurrence. An additional topic discussed was the description of an IDS and its types. The correct choice of the IDS type to develop depends on whether it is aimed at local or wide traffic detection, or even the detection of unknown anomalies by compromis-

ing accuracy. Anomaly-based detection is the most common IDS since it is dynamic and provides the detection of both known and unknown anomalies.

Many papers were reviewed in this study, with the purpose of providing a broad perspective of what has been done in anomaly detection, and what could be improved. From straightforward approaches to complex systems, there is a wide range of possibilities to address the anomaly detection problem, though every technique has its advantages and drawbacks. As stated in this survey, classification methods have the highest detection rates amongst all papers reviewed; however, they have some drawbacks regarding dependence on classification assumptions and resource consumption. Moreover, clustering-based techniques provide a stable performance in terms of detection rate and complexity, although they showed to be time-consuming and highly reliant on proximity measures. Ultimately, the literature cannot affirm which is the ideal technique to handle network anomalies, but a methodical investigation of each technique is essential to understand which problem domain it suits best.

Furthermore, this survey underlined the most relevant open issues within the field. The major gap observed is the unavailability of a standard and updated labeled dataset, so it would be a worthwhile investment to build some public database covering many anomalies and real traffic behaviors of distinct network infrastructure.

In conclusion, there are still some open issues to improve the effectiveness and feasibility of anomaly detection, but in contrast, there are also quite a few promising guidelines for researchers to follow in further investigations on the anomaly detection subject.

Acknowledgements This work was supported by Brazilian National Council for Scientific and Technological Development (CNPq) via Grant Nos. 249794/2013-6 and 309335/2017-5, and under Grant of Project 308348/2016-8; by National Funding from the FCT—*Fundação para a Ciência e a Tecnologia* through the UID/EEA/500008/2013 Project; by the Government of the Russian Federation, Grant 08-08; by Finep, with resources from Funttel, Grant No. 01.14.0231.00, under the Radiocommunication Reference Center (*Centro de Referência em Radiocomunicações*—CRR) project of the National Institute of Telecommunications (*Instituto Nacional de Telecomunicações*—Inatel), Brazil; and by the Research Center of the College of Computer and Information Sciences, King Saud University. The authors are grateful for this support.

References

1. Hashim, F., Munasinghe, K. S., & Jamalipour, A. (2010). Biologically inspired anomaly detection and security control frameworks for complex heterogeneous networks. *IEEE Transactions on Network and Service Management*, *7*, 268–281. https://doi.org/10.1109/TNSM.2010.1012.0360.
2. Xiao, X., Zhang, S., Mercaldo, F., Hu, G., & Sangaiah, A. K. (2017). Android malware detection based on system call sequences and LSTM. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-017-5104-0.
3. Balakrishnan, S. M., & Sangaiah, A. K. (2017). MIFIM—Middleware solution for service centric anomaly in future internet models. *Future Generation Computer Systems*, *74*, 349–365. https://doi.org/10.1016/j.future.2016.08.006.
4. Carvalho, L. F., Abrão, T., Mendes, L. S., & Proença, M. L. (2018). An ecosystem for anomaly detection and mitigation in software-defined networking. *Expert Systems with Applications*, *104*, 121–133. https://doi.org/10.1016/J.ESWA.2018.03.027.
5. Lu, S., Wang, X., & Mao, L. (2014). Network security situation awareness based on network simulation. In *2014 IEEE workshop on electronics, computer and applications* (pp. 512–517). https://doi.org/10.1109/IWECA.2014.6845671.
6. Hosseini Bamakan, S. M., Wang, H., & Shi, Y. (2017). Ramp loss K-support vector classification-regression: A robust and sparse multi-class approach to the intrusion detection problem. *Knowledge-Based Systems*, *126*, 113–126. https://doi.org/10.1016/j.knosys.2017.03.012.
7. Lof, A., & Nelson, R. (2014). Annotating network trace data for anomaly detection research. In *2014 IEEE 39th conference on local computer networks workshops (LCN workshops)* (pp. 679–684). https://doi.org/10.1109/LCNW.2014.6927720.
8. Barnett, V., & Lewis, T. (1994). *Outliers in statistical data* (3rd ed.). New York: Wiley.
9. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*, 1–58. https://doi.org/10.1145/1541880.1541882.
10. Lakhina, A., Crovella, M., & Diot, C. (2004). Diagnosing network-wide traffic anomalies. In *ACM SIGCOMM computer communication review* (Vol. 34, p. 219). https://doi.org/10.1145/1030194.1015492.
11. Hoque, N., Bhuyan, M. H., Baishya, R. C., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network attacks: Taxonomy, tools and systems. *Journal of Network and Computer Applications*, *40*, 307–324. https://doi.org/10.1016/j.jnca.2013.08.001.
12. Thottan, M., Liu, G., & Ji, C. (2010). Anomaly detection approaches for communication networks. In G. Cormode & M. Thottan (Eds.), *Algorithms for next generation networks* (pp. 239–261). London: Springer. https://doi.org/10.1007/978-1-84882-765-3_11.
13. Patcha, A., & Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, *51*, 3448–3470. https://doi.org/10.1016/j.comnet.2007.02.001.
14. Yu, Y. (2012). A survey of anomaly intrusion detection techniques. *Journal of Computing Sciences in Colleges*, *28*, 9–17.
15. Weiyu, Z., Qingbo, Y., & Yushui, G. (2009). A survey of anomaly detection methods in networks. In *International symposium on computer network and multimedia technology, 2009. CNMT 2009* (pp. 1–3). https://doi.org/10.1109/CNMT.2009.5374676.
16. Marnerides, A. K., Schaeffer-Filho, A., & Mauthe, A. (2014). Traffic anomaly diagnosis in Internet backbone networks: A survey. *Computer Networks*, *73*, 224–243. https://doi.org/10.1016/j.comnet.2014.08.007.
17. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys & Tutorials*, *16*, 303–336. https://doi.org/10.1109/SURV.2013.052213.00046.
18. Ahmed, M., Naser Mahmood, A., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, *60*, 19–31. https://doi.org/10.1016/j.jnca.2015.11.016.
19. Xiuyao, S., Mingxi, W., Jermaine, C., & Ranka, S. (2007). Conditional anomaly detection. *IEEE Transactions on Knowledge and*

Data Engineering, *19*, 631–644. https://doi.org/10.1109/TKDE.2007.1009.
20. Barford, P., Kline, J., Plonka, D., & Ron, A. (2002). A signal analysis of network traffic anomalies. In *Proceedings of the 2nd ACM SIGCOMM workshop on internet measurment—IMW '02*. ACM Press, New York, NY, USA (p. 71). https://doi.org/10.1145/637201.637210.
21. Barford, P., & Plonka, D. (2001). Characteristics of network traffic flow anomalies. *Proceedings of the 1st ACM SIGCOMM workshop on internet measurement* (pp. 69–73). https://doi.org/10.1145/505202.505211.
22. Jung, J., Krishnamurthy, B., & Rabinovich, M. (2002). Flash crowds and denial of service attacks. In *Proceedings of the 11th international conference on World Wide Web—WWW '02* (p. 293). https://doi.org/10.1145/511446.511485.
23. Pan, J., Hu, H., & Liu, Y. (2014). Human behavior during Flash Crowd in web surfing. *Physica A: Statistical Mechanics and Its Applications*, *413*, 212–219. https://doi.org/10.1016/j.physa.2014.06.085.
24. Ghorbani, A. A., Lu, W., & Tavallaee, M. (2010). Network attacks. *Advances in Information Security*, *47*, 1–25. https://doi.org/10.1007/978-0-387-88771-5_1.
25. Mouton, F., Malan, M. M., & Venter, H. S. (2013). Social engineering from a normative ethics perspective. In *Information security for South Africa, 2013* (pp. 1–8). https://doi.org/10.1109/ISSA.2013.6641064.
26. Maxion, R. A., & Townsend, T. N. (2002). Masquerade detection using truncated command lines. In *International conference on dependable systems and networks, 2002. DSN 2002. Proceedings* (pp. 219–228). https://doi.org/10.1109/DSN.2002.1028903.
27. Szor, P. (2005). *The art of computer virus research and defense*. Reading: Addison-Wesley.
28. Weaver, N., Paxson, V., Staniford, S., & Cunningham, R., (2003). A taxonomy of computer worms. In *Proceedings of the 2003 ACM workshop on Rapid malcode* (pp. 11–18). https://doi.org/10.1145/948187.948190.
29. Peng, T., Leckie, C., & Ramamohanarao, K. (2007). Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Computing Surveys*, *39*, 3. https://doi.org/10.1145/1216370.1216373.
30. Mohana Priya, P., Akilandeswari, V., Mercy Shalinie, S., Lavanya, V., & Shanmuga Priya, M. (2014). The protocol independent detection and classification (PIDC) system for DRDoS attack. In *2014 International conference on recent trends in information technology (ICRTIT)* (pp. 1–7). IEEE. https://doi.org/10.1109/ICRTIT.2014.6996154.
31. Muller, T., & Freiling, F. C. (2014). A systematic assessment of the security of full disk encryption. *IEEE Transactions on Dependable and Secure Computing*. https://doi.org/10.1109/TDSC.2014.2369041.
32. Raza, M., & Haider, W. (2012). A survey of password attacks and comparative analysis on methods for secure authentication. *World Applied Sciences Journal*, *19*, 439–444. https://doi.org/10.5829/idosi.wasj.2012.19.04.1837.
33. Shimonski, R., Zenir, J., & Bishop, A. (2015). Chapter 2: Information gathering. In R. S. Z. Bishop (Ed.), *Cyber Reconnaissance, surveillance and defense* (pp. 45–84). Boston: Syngress. https://doi.org/10.1016/B978-0-12-801308-3.00002-0.
34. Harrington, D., Presuhn, R., & Wijnen, B. (2002). RFC 3411: An architecture for describing simple network management protocol (SNMP) management frameworks (pp. 1–64). https://tools.ietf.org/html/rfc3411. Accessed 23 Oct 2017.
35. Thottan, M., & Ji, C. (2003). Anomaly detection in IP networks. *IEEE Transactions on Signal Processing*, *51*, 2191–2204. https://doi.org/10.1109/TSP.2003.814797.
36. Cabrera, J. B. D., Lewis, L., Qin, X., Lee, W., Prasanth, R. K., Ravichandran, B., & Mehra, R. K. (2001). Proactive detection of distributed denial of service attacks using MIB traffic variables: A feasibility study. In *2001 IEEE/IFIP International symposium on integrated network management proceedings. VII. Integr. Manag. Strateg. New Millenn. (Cat. No. 01EX470)* (pp. 609–622). IEEE. https://doi.org/10.1109/INM.2001.918069.
37. Yu, J., Lee, H., Kim, M.-S., & Park, D. (2008). Traffic flooding attack detection with SNMP MIB using SVM. *Computer Communications*, *31*, 4212–4219. https://doi.org/10.1016/j.comcom.2008.09.018.
38. Lima, M. F., Sampaio, L. D. H., Zarpelao, B. B., Rodrigues, J. J. P. C., Abrao, T., & Proenca, M. L., Jr. (2010). Networking anomaly detection using DSNs and particle swarm optimization with re-clustering. In *2010 IEEE global telecommunications conference GLOBECOM 2010* (pp. 1–6). IEEE. https://doi.org/10.1109/GLOCOM.2010.5683910.
39. Zarpelao, B. B., Mendes, L. S., Proenca Jr., M. L., & Rodrigues, J. J. P. C. (2009). Parameterized anomaly detection system with automatic configuration. In *GLOBECOM 2009—2009 IEEE global telecommunications conference* (pp. 1–6). IEEE. https://doi.org/10.1109/GLOCOM.2009.5426189.
40. Duffield, N., Haffner, P., Krishnamurthy, B., & Ringberg, H. (2009). Rule-based anomaly detection on IP flows. In *IEEE INFOCOM 2009—28th Conference on Computer Communications* (pp. 424–432). IEEE. https://doi.org/10.1109/INFCOM.2009.5061947.
41. Fontugne, R., & Fukuda, K. (2011). A Hough-transform-based anomaly detector with an adaptive time interval. *ACM SIGAPP Applied Computing Review*, *11*, 41–51. https://doi.org/10.1145/2034594.2034598.
42. Introduction to Cisco IOS®NetFlow (White Paper), (2012) 1–16. http://www.cisco.com/c/en/us/products/collateral/ios-nx-ossoftware/iosnetflow/prod_white_paper0900aecd80406232.pdf. Accessed 10 Dec 2017.
43. Claise, B. (2004). RFC 3954: Cisco systems netflow services export version 9 (pp. 1–33). https://tools.ietf.org/html/rfc3954. Accessed September 2, 2016.
44. Trammell, B., & Claise, B. (2013). RFC 7011: Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information, 1–53. 2070-1721.
45. Chapman, C. (2016). Chapter 10: Traffic performance testing in the network. In *Network performance and security* (pp. 295–317). https://doi.org/10.1016/B978-0-12-803584-9.00010-X.
46. NfSen: NetFlow sensor. (2011). http://nfsen.sourceforge.net/. Accessed September 2, 2016.
47. nTop. (2016). http://www.ntop.org/. Accessed September 2, 2016.
48. Panchen, S., Phaal, P., & McKee, N. (2001). RFC 3176: InMon Corporation's sFlow: A method for monitoring traffic in switched and routed networks, 1–31. https://tools.ietf.org/html/rfc3176. Accessed September 2, 2016.
49. Duffield, N. (2004). Sampling for passive internet measurement: A review. *Statistical Science*, *19*, 472–498. https://doi.org/10.1214/088342304000000206.
50. Cisco NetFlow-Lite Solution Overview, Cisco. (2016). http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-2960-x-series-switches/solution_overview_c22-728776.html. Accessed September 2, 2016.
51. Deri, L., Chou, E., Cherian, Z., Karmarkar, K., & Patterson, M. (2011). Increasing data center network visibility with cisco NetFlow-Lite. In *International conference on network and service management* (pp. 1–6).
52. Jadidi, Z., Muthukkumarasamy, V., Sithirasenan, E., & Singh, K. (2015). Flow-based anomaly detection in big data. In *Network big data* (pp. 257–279). Chapman and Hall/CRC. https://doi.org/10.1201/b18772-17.

53. Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., & Stiller, B. (2010). An overview of IP flow-based intrusion detection. *IEEE Communications Surveys and Tutorials*, *12*, 343–356. https://doi.org/10.1109/SURV.2010.032210.00054.
54. Winter, P., Hermann, E., & Zeilinger, M. (2011). Inductive intrusion detection in flow-based network data using one-class support vector machines. In *2011 4th IFIP international conference on new technologies, mobility and security* (pp. 1–5). IEEE. https://doi.org/10.1109/NTMS.2011.5720582.
55. Bartos, K., Rehak, M., & Krmicek, V. (2011). Optimizing flow sampling for network anomaly detection. In *2011 7th international wireless communications and mobile computing conference* (pp. 1304–1309). IEEE. https://doi.org/10.1109/IWCMC.2011.5982728.
56. Zhang, Y., Fang, B., & Luo, H. (2010). Identifying high-rate flows based on sequential sampling. *IEICE Transactions on Information and Systems*, *E93–D*, 1162–1174. https://doi.org/10.1587/transinf.E93.D.1162.
57. Silva, J. M. C., Carvalho, P., & Lima, S. R. (2015). Analysing traffic flows through sampling: A comparative study. In *2015 IEEE symposium on computers and communications* (pp. 341–346). https://doi.org/10.1109/ISCC.2015.7405538.
58. Kemmerer, R. A., & Vigna, G. (2002). Intrusion detection: A brief history and overview. *Computer*, *35*, 27–30. https://doi.org/10.1109/MC.2002.1012428.
59. Lee, W., & Stolfo, S. (1998). Data mining approaches for intrusion detection. In *Proceedings of 7th USENIX security symposium, USENIX Association* (pp. 6–6). https://dl.acm.org/citation.cfm?id=1267555. Accessed November 1, 2017.
60. Bul'ajoul, W., James, A., & Pannu, M. (2015). Improving network intrusion detection system performance through quality of service configuration and parallel technology. *Journal of Computer and System Sciences*, *81*, 981–999. https://doi.org/10.1016/j.jcss.2014.12.012.
61. Bostani, H., & Sheikhan, M. (2017). Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach. *Computer Communications*, *98*, 52–71. https://doi.org/10.1016/j.comcom.2016.12.001.
62. Milenkoski, A., Vieira, M., Kounev, S., Avritzer, A., & Payne, B. D. (2015). Evaluating computer intrusion detection systems: A survey of common practices. *ACM Computing Surveys*, *48*, 1–41. https://doi.org/10.1145/2808691.
63. Butun, I., Morgera, S. D., & Sankar, R. (2014). A survey of intrusion detection systems in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, *16*, 266–282. https://doi.org/10.1109/SURV.2013.050113.00191.
64. Debar, H., Dacier, M., & Wespi, A. (1999). Towards a taxonomy of intrusion–detection systems. *Computer Networks*, *31*, 805–822. https://doi.org/10.1016/S1389-1286(98)00017-6.
65. Meng Hui, L., & Jones, A. (2008). Network anomaly detection system: The state of art of network behaviour analysis. In *International conference on convergence and hybrid information technology 2008. ICHIT '08* (pp. 459–465). https://doi.org/10.1109/ICHIT.2008.249.
66. Sobh, T. S. (2006). Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art. *Computer Standards & Interfaces*, *28*, 670–694. https://doi.org/10.1016/j.csi.2005.07.002.
67. de Assis, M. V. O., Rodrigues, J. J. P. C., & Proença, M. L. (2014). A seven-dimensional flow analysis to help autonomous network management. *Information Sciences*, *278*, 900–913. https://doi.org/10.1016/j.ins.2014.03.102.
68. Stakhanova, N., Basu, S., & Wong, J. (2010). On the symbiosis of specification-based and anomaly-based detection. *Computers & Security*, *29*, 253–268. https://doi.org/10.1016/j.cose.2009.08.007.
69. Lakhina, A., Papagiannaki, K., Crovella, M., Diot, C., Kolaczyk, E. D., & Taft, N. (2004). Structural analysis of network traffic flows. *ACM SIGMETRICS Performance Evaluation Review*, *32*, 61. https://doi.org/10.1145/1012888.1005697.
70. Lakhina, A., Crovella, M., & Diot, C. (2005). Mining anomalies using traffic feature distributions. *ACM SIGCOMM Computer Communication Review*, *35*, 217. https://doi.org/10.1145/1090191.1080118.
71. Callegari, C., Giordano, S., Pagano, M., & Pepe, T. (2011). Combining sketches and wavelet analysis for multi time-scale network anomaly detection. *Computers & Security*, *30*, 692–704. https://doi.org/10.1016/j.cose.2011.08.006.
72. Hamdi, M., & Boudriga, N. (2007). Detecting Denial-of-Service attacks using the wavelet transform. *Computer Communications*, *30*, 3203–3213. https://doi.org/10.1016/j.comcom.2007.05.061.
73. Jolliffe, I. T. (2002). *Principal component analysis*. Berlin: Springer.
74. Jackson, J. E. (2005). *A user's guide to principal components*. New York: Wiley.
75. Ringberg, H., Soule, A., Rexford, J., & Diot, C. (2007). Sensitivity of PCA for traffic anomaly detection. *SIGMETRICS Performance Evaluation Review*, *35*, 109–120. https://doi.org/10.1145/1269899.1254895.
76. Wright, J., Ganesh, A., Rao, S., Peng, Y., & Ma, Y. (2009). Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *Advances in neural information processing systems* (Vol. 22, pp. 2080–2088). Curran Associates, Inc. http://papers.nips.cc/paper/3704-robust-principal-component-analysis-exact-recovery-of-corrupted-low-rank-matrices-via-convex-optimization.pdf.
77. Candès, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? *Journal of the ACM*, *58*, 11:1–11:37. https://doi.org/10.1145/1970392.1970395.
78. Pascoal, C., Rosario de Oliveira, M., Valadas, R., Filzmoser, P., Salvador, P., & Pacheco, A. (2012). Robust feature selection and robust PCA for internet traffic anomaly detection. In *INFOCOM, 2012 Proceedings of IEEE* (pp. 1755–1763). https://doi.org/10.1109/INFCOM.2012.6195548.
79. Kanda, Y., Fontugne, R., Fukuda, K., & Sugawara, T. (2013). ADMIRE: Anomaly detection method using entropy-based PCA with three-step sketches. *Computer Communications*, *36*, 575–588. https://doi.org/10.1016/j.comcom.2012.12.002.
80. OReilly, C., Gluhak, A., & Imran, M. A. (2016). Distributed anomaly detection using minimum volume elliptical principal component analysis. *IEEE Transactions on Knowledge and Data Engineering*, *28*, 2320–2333. https://doi.org/10.1109/TKDE.2016.2555804.
81. Camacho, J., Pérez-Villegas, A., García-Teodoro, P., & Maciá-Fernández, G. (2016). PCA-based multivariate statistical network monitoring for anomaly detection. *Computers & Security*, *59*, 118–137. https://doi.org/10.1016/j.cose.2016.02.008.
82. Fernandes, G., Carvalho, L. F., Rodrigues, J. J. P. C., & Proença, M. L. (2016). Network anomaly detection using IP flows with principal component analysis and ant colony optimization. *Journal of Network and Computer Applications*, *64*, 1–11. https://doi.org/10.1016/j.jnca.2015.11.024.
83. Fernandes, G., Rodrigues, J. J. P. C., & Proença, M. L. (2015). Autonomous profile-based anomaly detection system using principal component analysis and flow analysis. *Applied Soft Computing*, *34*, 513–525. https://doi.org/10.1016/j.asoc.2015.05.019.
84. Fernandes, G., Zacaron, A. M., Rodrigues, J. J. P. C., & Proenca, M. L. (2013). Digital signature to help network management using principal component analysis and K-means clustering. In *2013 IEEE international conference on communications* (pp. 2519–2523). IEEE. https://doi.org/10.1109/ICC.2013.6654912.

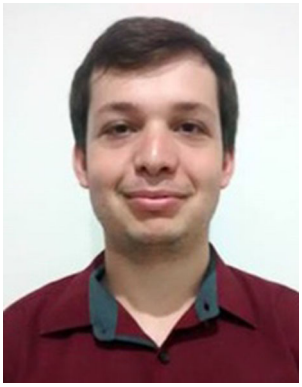
85. Yeung, D. S., Shuyuan, J., & Xizhao, W. (2007). Covariance-matrix modeling and detecting various flooding attacks. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *37*, 157–169. https://doi.org/10.1109/TSMCA.2006.889480.
86. Xie, M., Hu, J., & Guo, S. (2015). Segment-based anomaly detection with approximated sample covariance matrix in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, *26*, 574–583. https://doi.org/10.1109/TPDS.2014.2308198.
87. Huang, T., Sethu, H., & Kandasamy, N. (2016). A new approach to dimensionality reduction for anomaly detection in data traffic. *IEEE Transactions on Network and Service Management*, *13*, 651–665. https://doi.org/10.1109/TNSM.2016.2597125.
88. Kalkan, K., & Alagöz, F. (2016). A distributed filtering mechanism against DDoS attacks: ScoreForCore. *Computer Networks*, *108*, 199–209. https://doi.org/10.1016/j.comnet.2016.08.023.
89. Ozkan, H., Ozkan, F., & Kozat, S. S. (2016). Online anomaly detection under Markov statistics with controllable type-I error. *IEEE Transactions on Signal Processing*, *64*, 1435–1445. https://doi.org/10.1109/TSP.2015.2504345.
90. Proença, M. L., Coppelmans, C., Bottoli, M., Alberti, A., & Mendes, L. S. (2004). The hurst parameter for digital signature of network segment. In J. N. de Souza, P. Dini, & P. Lorenz (Eds.), *Telecommunications and networking—ICT 2004 11th international conference on telecommunications, Fortaleza, Brazil, August 1–6, 2004. Proceedings* (pp. 772–781). Berlin: Springer. https://doi.org/10.1007/978-3-540-27824-5_103.
91. Pena, E. H. M., Carvalho, L. F., Barbon, S, Jr., Rodrigues, J. J. P. C., & Proença, M. L, Jr. (2017). Anomaly detection using the correlational paraconsistent machine with digital signatures of network segment. *Information Sciences*, *420*, 313–328. https://doi.org/10.1016/j.ins.2017.08.074.
92. Pena, E. H. M., Carvalho, L. F., Barbon, S., Rodrigues, J. J. P. C., & Proenca, M. L. (2014). Correlational paraconsistent machine for anomaly detection. In *2014 IEEE global communications conference* (pp. 551–556). IEEE. https://doi.org/10.1109/GLOCOM.2014.7036865.
93. Bang, J., Cho, Y.-J., & Kang, K. (2017). Anomaly detection of network-initiated LTE signaling traffic in wireless sensor and actuator networks based on a Hidden semi-Markov Model. *Computers & Security*, *65*, 108–120. https://doi.org/10.1016/j.cose.2016.11.008.
94. Ren, H., Ye, Z., & Li, Z. (2017). Anomaly detection based on a dynamic Markov model. *Computers & Security*. https://doi.org/10.1016/j.ins.2017.05.021.
95. Jazi, H. H., Gonzalez, H., Stakhanova, N., & Ghorbani, A. A. (2017). Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. *Computer Networks*, *121*, 25–36. https://doi.org/10.1016/j.comnet.2017.03.018.
96. Han, J., Kamber, M., & Pei, J. (2012). 10: Cluster analysis: Basic concepts and methods. In J. H. Kamber, & J. Pei (Eds.), *Data mininig* (3d edn., pp. 443–495). Boston: Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-381479-1.00010-1.
97. Rajasegarar, S., Leckie, C., & Palaniswami, M. (2014). Hyperspherical cluster based distributed anomaly detection in wireless sensor networks. *Journal of Parallel and Distributed Computing*, *74*, 1833–1847. https://doi.org/10.1016/j.jpdc.2013.09.005.
98. Mazel, J., Casas, P., Labit, Y., & Owezarski, P. (2011). Sub-space clustering, inter-clustering results association and anomaly correlation for unsupervised network anomaly detection. In *CNSM '11 Proceedings of the 7th international conference on network and services management* (pp. 73–80). http://dl.acm.org/citation.cfm?id=2147683.
99. Karami, A., & Guerrero-Zapata, M. A. (2015). Fuzzy anomaly detection system based on hybrid PSO-K means algorithm in content-centric networks. *Neurocomputing*, *149*, 1253–1269. https://doi.org/10.1016/j.neucom.2014.08.070.
100. Carvalho, L. F., Barbon, S., Mendes, L. S., & Proença, M. L. (2016). Unsupervised learning clustering and self-organized agents applied to help network management. *Expert Systems with Applications*, *54*, 29–47. https://doi.org/10.1016/j.eswa.2016.01.032.
101. Dromard, J., Roudiere, G., & Owezarski, P. (2017). Online and scalable unsupervised network anomaly detection method. *IEEE Transactions on Network and Service Management*, *14*, 34–47. https://doi.org/10.1109/TNSM.2016.2627340.
102. He, D., Chan, S., Ni, X., & Guizani, M. (2017). Software-defined-networking-enabled traffic anomaly detection and mitigation. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2017.2694702.
103. Bigdeli, E., Mohammadi, M., Raahemi, B., & Matwin, S. (2018). Incremental anomaly detection using two-layer cluster-based structure. *Information Sciences*, *429*, 315–331. https://doi.org/10.1016/j.ins.2017.11.023.
104. Estevez-Tapiador, J. M., Garcia-Teodoro, P., & Diaz-Verdejo, J. E. (2003). Stochastic protocol modeling for anomaly based network intrusion detection. In *Information assurance. 2003. IWIAS 2003. Proceedings. First IEEE international workshop on* (pp. 3–12). https://doi.org/10.1109/IWIAS.2003.1192454.
105. Su, M.-Y. (2010). Discovery and prevention of attack episodes by frequent episodes mining and finite state machines. *Journal of Network and Computer Applications*, *33*, 156–167. https://doi.org/10.1016/j.jnca.2009.10.003.
106. Hammerschmidt, C., Marchal, S., State, R., Pellegrino, G.,& Verwer, S., (2016). Efficient learning of communication profiles from IP flow records. In *2016 IEEE 41st conference on local computer networks* (pp. 559–562). IEEE. https://doi.org/10.1109/LCN.2016.92.
107. Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*. New York: Wiley.
108. Klassen, M., & Ning, Y. (2012). Anomaly based intrusion detection in wireless networks using Bayesian classifier. In *2012 IEEE fifth international conference on advanced computational intelligence (ICACI)* (pp. 257–264). https://doi.org/10.1109/ICACI.2012.6463163.
109. Tao, L., Ailing, Q., Yuanbin, H., & Xintan, C. (2008). Method for network anomaly detection based on Bayesian statistical model with time slicing. In *7th world congress on intelligent control and automation, 2008. WCICA 2008* (pp. 3359–3362). https://doi.org/10.1109/WCICA.2008.4593458.
110. Swarnkar, M., & Hubballi, N. (2016). OCPAD: One class Naive Bayes classifier for payload based anomaly detection. *Expert Systems with Applications*, *64*, 330–339. https://doi.org/10.1016/j.eswa.2016.07.036.
111. Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.
112. Catania, C. A., Bromberg, F., & Garino, C. G. (2012). An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection. *Expert Systems with Applications*, *39*, 1822–1829. https://doi.org/10.1016/j.eswa.2011.08.068.
113. Amer, M., Goldstein, M., & Abdennadher, S. (2013). Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD workshop on outlier detection and description* (pp. 8–15). https://doi.org/10.1145/2500853.2500857.
114. Erfani, S. M., Rajasegarar, S., Karunasekera, S., & Leckie, C. (2016). High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, *58*, 121–134. https://doi.org/10.1016/j.patcog.2016.03.028.

115. Wang, H., Gu, J., & Wang, S. (2017). An effective intrusion detection framework based on SVM with feature augmentation. *Knowledge-Based Systems*, *136*, 130–139. https://doi.org/10.1016/j.knosys.2017.09.014.
116. Kabir, E., Hu, J., Wang, H., & Zhuo, G. (2017). A novel statistical technique for intrusion detection systems. *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2017.01.029.
117. Subba, B., Biswas, S., & Karmakar, S. (2016). A neural network based system for intrusion detection and attack classification. In *2016 Twenty second national conference on communication* (pp. 1–6). IEEE. https://doi.org/10.1109/NCC.2016.7561088.
118. Saeed, A., Ahmadinia, A., Javed, A., & Larijani, H. (2016). Intelligent intrusion detection in low-power IoTs. *ACM Transactions on Internet Technology*, *16*, 1–25. https://doi.org/10.1145/2990499.
119. Brown, J., Anwar, M., & Dozier, G. (2016). An evolutionary general regression neural network classifier for intrusion detection. In *2016 25th International conference on computer communication and networks (ICCCN)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICCCN.2016.7568493.
120. Ashfaq, R. A. R., Wang, X.-Z., Huang, J. Z., Abbas, H., & He, Y.-L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences*, *378*, 484–497. https://doi.org/10.1016/j.ins.2016.04.019.
121. Peddabachigari, S., Abraham, A., Grosan, C., & Thomas, J. (2007). Modeling intrusion detection system using hybrid intelligent systems. *Journal of Network and Computer Applications*, *30*, 114–132. https://doi.org/10.1016/j.jnca.2005.06.003.
122. Aburomman, A. A., & Bin Ibne Reaz, M. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing*, *38*, 360–372. https://doi.org/10.1016/j.asoc.2015.10.011.
123. Sornsuwit, P., & Jaiyen, S. (2015). Intrusion detection model based on ensemble learning for U2R and R2L attacks. In *2015 7th international conference on information technology and electrical engineering (ICITEE)* (pp. 354–359). IEEE. https://doi.org/10.1109/ICITEED.2015.7408971.
124. Bukhtoyarov, V., & Zhukov, V. (2014). Ensemble-distributed approach in classification problem solution for intrusion detection systems. In E. Corchado, J. A. Lozano, H. Quintián, & H. Yin (Eds.), *2014 15th International conference on intelligent data engineering automated learning—IDEAL*, Salamanca, Spain, September 10–12, 2014. Proceedings (pp. 255–265). Cham: Springer. https://doi.org/10.1007/978-3-319-10840-7_32.
125. Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*, 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.
126. Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). New York: Wiley.
127. Lee, W., & Xiang, D. (2001). Information-theoretic measures for anomaly detection. In *Proceedings of 2001 IEEE symposium on security and privacy, S&P 2001* (pp. 130–143). IEEE Comput. Soc, n.d. https://doi.org/10.1109/SECPRI.2001.924294.
128. David, J., & Thomas, C. (2015). DDoS attack detection using fast entropy approach on flow- based network traffic. *Procedia Computer Science*, *50*, 30–36. https://doi.org/10.1016/j.procs.2015.04.007.
129. Amaral, A. A., Mendes, L. S., Zarpelão, B. B., & Junior, M. L. P. (2017). Deep IP flow inspection to detect beyond network anomalies. *Computer Communications*, *98*, 80–96. https://doi.org/10.1016/j.comcom.2016.12.007.
130. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2016). A multi-step outlier-based anomaly detection approach to network-wide traffic. *Information Sciences*, *348*, 243–271. https://doi.org/10.1016/j.ins.2016.02.023.
131. Bereziński, P., Jasiul, B., & Szpyrka, M. (2015). An entropy-based network anomaly detection method. *Entropy*, *17*, 2367–2408. https://doi.org/10.3390/e17042367.
132. Behal, S., & Kumar, K. (2017). Detection of DDoS attacks and flash events using novel information theory metrics. *Computer Networks*, *116*, 96–110. https://doi.org/10.1016/j.comnet.2017.02.015.
133. Xie, M., Hu, J., Guo, S., & Zomaya, A. Y. (2017). Distributed segment-based anomaly detection with Kullback–Leibler divergence in wireless sensor networks. *IEEE Transactions on Information Forensics and Security*, *12*, 101–110. https://doi.org/10.1109/TIFS.2016.2603961.
134. Li, G., & Wang, Y. (2012). Differential Kullback–Leibler divergence based anomaly detection scheme in sensor networks. In *2012 IEEE 12th international conference on computer and information technology* (pp. 966–970). IEEE. https://doi.org/10.1109/CIT.2012.197.
135. Kar, A. K. (2016). Bio inspired computing: A review of algorithms and scope of applications. *Expert Systems with Applications*, *59*, 20–32. https://doi.org/10.1016/j.eswa.2016.04.018.
136. Firdaus, A., Anuar, N. B., Razak, M. F. A., & Sangaiah, A. K. (2017). Bio-inspired computational paradigm for feature investigation and malware detection: Interactive analytics. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-017-4586-0.
137. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, *18*, 1153–1176. https://doi.org/10.1109/COMST.2015.2494502.
138. Sen, S. (2015). A survey of intrusion detection systems using evolutionary computation. In *Bio-inspired computation in telecommunications* (pp. 73–94). Elsevier. https://doi.org/10.1016/B978-0-12-801538-4.00004-5.
139. de Castro, L. N., & Timmis, J. (2002). *Artificial immune systems: A new computational intelligence approach*. London: Springer.
140. Saurabh, P., & Verma, B. (2016). An efficient proactive artificial immune system based anomaly detection and prevention system. *Expert Systems with Applications*, *60*, 311–320. https://doi.org/10.1016/j.eswa.2016.03.042.
141. Igbe, O., Darwish, I., & Saadawi, T. (2016). Distributed network intrusion detection systems: An artificial immune system approach. In *2016 IEEE First international conference on connected health: applications, systems and engineering technologies* (pp. 101–106). IEEE. https://doi.org/10.1109/CHASE.2016.36.
142. Shamshirband, S., Anuar, N. B., Kiah, M. L. M., Rohani, V. A., Petković, D., Misra, S., et al. (2014). Co-FAIS: Cooperative fuzzy artificial immune system for detecting intrusion in wireless sensor networks. *Journal of Network and Computer Applications*, *42*, 102–117. https://doi.org/10.1016/j.jnca.2014.03.012.
143. Aslahi-Shahri, B. M., Rahmani, R., Chizari, M., Maralani, A., Eslami, M., Golkar, M. J., et al. (2016). A hybrid method consisting of GA and SVM for intrusion detection system. *Neural Computing and Applications*, *27*, 1669–1676. https://doi.org/10.1007/s00521-015-1964-2.
144. Singh, S., & Kushwah, R. S. (2016). Energy efficient approach for intrusion detection system for WSN by applying optimal clustering and genetic algorithm. In *Proceedings of the international conference on advances in information communication technology & computing—AICTC '16* (pp. 1–6). New York, NY: ACM Press. https://doi.org/10.1145/2979779.2979840.
145. Hamamoto, A. H., Carvalho, L. F., Sampaio, L. D. H., Abrão, T., & Proença, M. L. (2018). Network anomaly detection system using genetic algorithm and fuzzy logic. *Expert Systems with Applications*, *92*, 390–402. https://doi.org/10.1016/j.eswa.2017.09.013.

146. Elsayed, S., Sarker, R., & Slay, J. (2015). Evaluating the performance of a differential evolution algorithm in anomaly detection. In *2015 IEEE congress on evolutionary computation* (pp. 2490–2497). IEEE. https://doi.org/10.1109/CEC.2015.7257194.
147. Huang, C.-L., & Dun, J.-F. (2008). A distributed PSO-SVM hybrid system with feature selection and parameter optimization. *Applied Soft Computing*, *8*, 1381–1391. https://doi.org/10.1016/j.asoc.2007.10.007.
148. Lin, S.-W., Ying, K.-C., Chen, S.-C., & Lee, Z.-J. (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications*, *35*, 1817–1824. https://doi.org/10.1016/j.eswa.2007.08.088.
149. Hosseini Bamakan, S. M., Wang, H., Yingjie, T., & Shi, Y. (2016). An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. *Neurocomputing*, *199*, 90–102. https://doi.org/10.1016/j.neucom.2016.03.031.
150. de Assis, M. V. O., Hamamoto, A. H., Abrao, T., & Proenca, M. L. (2017). A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for DoS/DDoS mitigation on SDN networks. *IEEE Access*. https://doi.org/10.1109/ACCESS.2017.2702341.
151. Grill, M., & Pevný, T. (2016). Learning combination of anomaly detectors for security domain. *Computer Networks*, *107*, 55–63. https://doi.org/10.1016/j.comnet.2016.05.021.
152. Al-Yaseen, W. L., Othman, Z. A., & Nazri, M. Z. A. (2017). Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Systems with Applications*, *67*, 296–303. https://doi.org/10.1016/j.eswa.2016.09.041.
153. Forestiero, A. (2016). Self-organizing anomaly detection in data streams. *Information Sciences*, *373*, 321–336. https://doi.org/10.1016/j.ins.2016.09.007.
154. Salem, O., Guerassimov, A., Mehaoua, A., Marcus, A., & Furht, B. (2014). Anomaly detection in medical wireless sensor networks using SVM and linear regression models. *International Journal of E-Health and Medical Communications*, *5*, 20–45. https://doi.org/10.4018/ijehmc.2014010102.
155. Wang, W., Liu, J., Pitsilis, G., & Zhang, X. (2016). Abstracting massive data for lightweight intrusion detection in computer networks. *Information Sciences*. https://doi.org/10.1016/j.ins.2016.10.023.
156. Adaniya, M. H. A. C., Abrão, T., & Proença, M. L, Jr. (2013). Anomaly detection using metaheuristic firefly harmonic clustering. *Journal of Networks*, *8*, 82–91. https://doi.org/10.4304/jnw.8.1.82-91.
157. Proenca, M. L., Zarpelao, B. B., & Mendes, L. S. (2005). Anomaly detection for network servers using digital signature of network segment. In *Advanced industrial conference on telecommunications/service assurance with partial and intermittent resources conference/e-learning on telecommunications workshop* (pp. 290–295). IEEE. https://doi.org/10.1109/AICT.2005.26.
158. Chen, M.-H., Chang, P.-C., & Wu, J.-L. (2016). A population-based incremental learning approach with artificial immune system for network intrusion detection. *Engineering Applications of Artificial Intelligence*, *51*, 171–181. https://doi.org/10.1016/j.engappai.2016.01.020.
159. Grill, M., Pevný, T., & Rehak, M. (2017). Reducing false positives of network anomaly detection by local adaptive multivariate smoothing. *Journal of Computer and System Sciences*, *83*, 43–57. https://doi.org/10.1016/j.jcss.2016.03.007.
160. Guo, C., Ping, Y., Liu, N., & Luo, S.-S. (2016). A two-level hybrid approach for intrusion detection. *Neurocomputing*, *214*, 391–400. https://doi.org/10.1016/j.neucom.2016.06.021.

Gilberto Fernandes Jr. is a Ph.D. student on computer networks at University of Beira Interior/ Instituto de Telecomunicações, Covilhã, Portugal. He received the title of MSc degree in Computer Science from the Computer Science Department at State University of Londrina (UEL), Brazil, in 2014. He is a member of the NetGNA Research Group (http://netgna.it.ubi.pt) supervised by Joel J. P. C. Rodrigues. His research interests include computer networks, network operation, network management, network security and e-Health systems.

Joel J. P. C. Rodrigues (joeljr@ieee.org) [S'01, M'06, SM'06] is a professor and senior researcher at the National Institute of Telecommunications (Inatel), Brazil and senior researcher at the Instituto de Telecomunicações, Portugal. He has been professor at the University of Beira Interior (UBI), Portugal and visiting professor at the University of Fortaleza (UNIFOR), Brazil. He received the Academic Title of Aggregated Professor in informatics engineering from UBI, the Habilitation in computer science and engineering from the University of Haute Alsace, France, a Ph.D. degree in informatics engineering and an M.Sc. degree from the UBI, and a five-year B.Sc. degree (licentiate) in informatics engineering from the University of Coimbra, Portugal. His main research interests include e-health, sensor networks and IoT, vehicular communications, and mobile and ubiquitous computing. Prof. Rodrigues is the leader of the Internet of Things research group (CNPq) and of the NetGNA Research Group, Director for Conference Development—IEEE ComSoc Board of Governors, IEEE Distinguished Lecturer, the President of the scientific council at ParkUrbis—Covilhã Science and Technology Park, the Past-Chair of the IEEE ComSoc Technical Committee on eHealth, the Past-chair of the IEEE ComSoc Technical Committee on Communications Software, Steering Committee member of the IEEE Life Sciences Technical Community and Publications co-Chair, and Member Representative of the IEEE Communications Society on the IEEE Biometrics Council. He is the editor-in-chief of the International Journal on E-Health and Medical Communications, the editor-in-chief of the Recent Advances on Communications and Networking Technology, the editor-in-chief of the Journal of Multimedia Information Systems, and editorial board member of several high-reputed journals. He has been general chair and TPC Chair of many international conferences, including IEEE ICC, IEEE GLOBECOM, and IEEE HEALTHCOM. He is a member of many international TPCs and participated in several international conferences organization. He has authored or coauthored over 600 papers in refereed international journals and conferences, 3 books, and 2 patents. He had been awarded several Outstanding Leadership and Outstanding Service Awards by IEEE Communications Society and several best papers awards. Prof. Rodrigues is a licensed professional engineer (as senior member), member of the Internet Society, and a senior member ACM and IEEE.



Luiz Fernando Carvalho is a Ph.D. candidate in Electrical Engineering and Telecommunications at State University of Campinas. He completed his master degree in Computer Science at State University of Londrina in 2014. Currently, he is a lecturer and member of the Computer Networks and Data Communication research group at State University of Londrina. His main research interests are management and security of computer networks and Software-defined Networking.

Jalal F. Al-Muhtadi PhD, is the Director of the Center of Excellence in Information Assurance (CoEIA) at King Saud University. He is also an Assistant Professor at the department of Computer Science at King Saud University. Areas of expertise include cybersecurity, information assurance, privacy, and Internet of Things. He received his PhD and MS degrees in Computer Science from the University of Illinois at Urbana-Champaign, USA. He has over 50 scientific publications in the areas of cybersecurity and IoT.

Mario Lemes Proença Jr. is an Associate Professor and leader of the research group that studies computer networks in the Computer Science Department at State University of Londrina (UEL), Brazil. He received the Ph.D. degree in Electrical Engineering and Telecommunications from State University of Campinas (UNICAMP) in 2005. He received the title of M.Sc degree in Computer Science from the Informatics Institute of Federal University of Rio Grande do Sul (UFRGS), in 1998. He has authored or coauthored over 100 papers in refereed international journals and conferences, books chapters, and one software register patent. His research interests include Computer Network, Network Operations, Management and Security and IT Governance. He has supervised 12 M.Sc. and two Ph.D. students. He has been a Master's supervisor at computer science in State University of Londrina and Ph.D. supervisor in Department of Electrical Engineering at UEL.