Available online at www.sciencedirect.com

SciVerse ScienceDirect

Journal of Information Security and Applications 00 (2019) 000–000

# Computational Intelligence Intrusion Detection Techniques in Mobile Cloud Computing Environments: Review, Taxonomy, and Open Research Issues

Shahab Shamshirband[1]*, Mahdis Fathi[2], Anthony T. Chronopoulos[3,4], Antonio Montieri[5], Fabio Palumbo[5], Antonio Pescapè[5]*

[1]*Department of Computer Science, Norwegian University Science and Technology, Trondheim, Norway*
[2]*Department of Computer Science, Islamiz Azad University, Iran*
[3]*Department of Computer Science, University of Texas at San Antonio, USA*
[4]*(Visiting Faculty) Dept of Computer Engineering & Informatics, University of Patras, Greece*
[6] *Electrical Engineering and Information Technology Department, University of Napoli Federico II, Italy*

**Abstract**

With the increasing utilization of the Internet and its provided services, an increase in cyber-attacks to exploit the information occurs. A technology to store and maintain user's information that is mostly used for its simplicity and low-cost services is cloud computing (CC). Also, a new model of computing that is noteworthy today is mobile cloud computing (MCC) that is used to reduce the limitations of mobile devices by allowing them to offload certain computations to the remote cloud. The cloud environment may consist of critical or essential information of an organization; therefore, to prevent this environment from possible attacks a security solution is needed. An intrusion detection system (IDS) is a solution to these security issues. An IDS is a hardware or software device that can examine all inside and outside network activities and recognize doubtful patterns that may demonstrate a network attack and automatically alert the network (or system) administrator. Because of the ability of an IDS to detect known/unknown (inside/outside) attacks, it is an excellent choice for securing cloud computing. Various methods are used in an intrusion detection system to recognize attacks more accurately. Unlike survey papers presented so far, this paper aims to present a comprehensive survey of intrusion detection systems that use computational intelligence (CI) methods in a (mobile) cloud environment. We firstly provide an overview of CC and MCC paradigms and service models, also reviewing security threats in these contexts. Previous literature is critically surveyed, highlighting the advantages and limitations of previous work. Then we define a taxonomy for IDS and classify CI-based techniques into single and hybrid methods. Finally, we highlight open issues and future directions for research on this topic.



* Corresponding authors. Tel.: +39 081 7683856; fax: +39 081 7683816.
*E-mail addresses:* pescape@unina.it; shahab.shamshirband@ntnu.no.

*Keywords*: Cloud Computing; Computational Intelligence; Intrusion Detection System; Mobile Cloud Computing; Security.

***Nomenclatures***

| | | | |
|---|---|---|---|
| Artificial Immune Systems | AIS | Intrusion Prevention System | IPS |
| Application-based Intrusion Detection System | AIDS | International Data Corporation | IDC |
| Address Resolution Protocol | ARP | Infrastructure as a Service | IaaS |
| Artificial Neural Networks | ANN | Intrusion Detection Systems | IDS |
| Cloud Computing | CC | Local Outlier Factor | LOF |
| Center for Applied Internet Data Analysis | CAIDA | Mean Absolute Error | MAE |
| Computational Intelligence | CI | National Institute of Standards and Technology | NIST |
| Denial of Service | DoS | Network-based Intrusion Detection System | NIDS |
| Decision Tree | DT | Network Behavior Analysis | NBA |
| (Distributed) Denial of Service | DDOS | Neural Network | NN |
| Distributed Intrusion Detection System | DIDS | Platform as a Service | PaaS |
| Evolutionary Computation | EC | Root Mean Square Error | RMSE |
| Elastic Compute Cloud | EC2 | Particle Swarm Optimization | PSO |
| Fuzzy Associative Pattern-based | FAP | Software as a Service | SaaS |
| Fuzzy Association Rule-based | FAR | Swarm Intelligence | SI |
| Genetic Algorithm | GA | Support Vector Machine | SVM |
| Game Theory | GT | Virtual Machine | VM |
| Host-based Intrusion Detection System | HIDS | Virtual Machine Introspection | VMI |
| Hypervisor Introspection | HVI | Virtual Machine Monitor | VMM |
| Hypervisor-based Intrusion Detection System | HVIDS | Wireless-based Intrusion Detection System | WIDS |

## 1. Introduction

Cloud Computing (CC) provides on-demand network access to a set of configurable computing resources such as services, servers, networks, applications, and storage which can be released rapidly with less service provider interactions or management endeavors. Based on the definition of NIST [1], different types of services, capabilities, and resources provided to the users are defined for CC, leading to three different service models: SaaS, PaaS, and IaaS.

Reliability and convenience are the two main reasons to use CC. Since cloud services are provided through the Internet, security, and privacy of these services must be considered. According to a study from the International Data Corporation [3], security is an essential challenge in the CC environment. Indeed, attacks such as ARP spoofing, DoS, Distributed DoS, Flooding, DNS poisoning, IP spoofing, etc. can occur in CC.

IDSs and IPSs are approaches that can alleviate the above attacks. An IDS is a hardware or software device that automates the process carried out to detect possible intrusions. An IPS is a software or hardware device that has all the capabilities of an IDS and can also stop the probable incidents. An IPS can reply to a detected threat or can change the security environment. It can change other security controls to stop the attack. The

main difference between IDS and IPS is that the IDS is a monitoring system, whereas the IPS is a control system that prevents the packet from delivery based on the contents of the packet [4].

Unfortunately, the latter is not a simple task to accomplish. When complex problems could not be solved by the traditional modeling, a collection of nature or human-inspired computational methods called Computational Intelligence (CI) are often used [5]. ANN, EC, SI, AIS, and Fuzzy Systems are examples of the CI paradigm [6]. Using CI techniques for IDS can eliminate the problem of distinguishing between abnormal and normal activities.

With the growing usage of mobile devices in everyday life [132], Mobile Cloud Computing (MCC) has also risen as a paradigm to cope with the limited computational capabilities and energy constraints of mobile devices, by allowing to offload computation to the remote cloud [13]. It is worth noting that the concept of Mobile Edge Computing (MEC) is a distinct one, although being derived from MCC, pushing the computation closer (i.e. at the network edge) to the mobile device than MCC and thus implying different solutions and challenges being out of the scope of this work. Indeed, in the following we focus on IDS in MCC only, leaving for future research the investigation of these issues in MEC and Fog computing scenarios.

Several surveys in the field of CC/MCC have been presented until now. Since security and privacy are two critical factors today [156, 157, 191], some of the survey papers address studies based on attacks, security and intrusion detection, and prevention techniques. Patel et al. [4] surveyed literature about intrusion detection and prevention techniques and classified them based on the layered architecture of CC. The authors explained some challenges for the development of intrusion detection and prevention techniques in the CC environment, and they also provided a list of requirements for cloud-based intrusion detection and prevention systems. In [3], the authors have classified IDS and IPS techniques for CC and have discussed several open security challenges in this field. Also, they briefly reviewed the types of firewall proposed for different kinds of attacks. C. N. Modi and K. Acha [7] studied the vulnerabilities and attacks on the virtualization layer of CC, taking into account also some related IPSs and IDSs papers. Furthermore, authors classified necessities and issues of cloud IDSs. Another survey on CC security is presented in [8]. In this survey, CC attacks were classified based on cloud models, and several defense mechanisms related articles were studied. Starting from this analysis, the authors provided challenges and open security issues of CC. Osanaiye et al. [9] surveyed Distributed Denial of Service (DDoS) attack related articles and defense mechanisms. Furthermore, they described each defense mechanism in detail with their advantages and disadvantages, providing open issues and some future directions. Mishra et al. [10] also presented a survey based on IDS techniques for the cloud environment. They studied existing attacks and threats in the cloud and discussed the IDS solutions and their benefits and limitations. At last, they mentioned some of the existing issues of CC. Authors in [11] discussed traditional attacks such as Flooding Attacks, Economic Denial of Sustainability (EDoS) Attacks, User to Root Attacks, Port Scanning Attacks, Backdoor Channel Attacks, Attacks on Virtual Machines or Hypervisors in the cloud environment. They investigated intelligent IDS such as SVM-based, anomaly-based, associated, GA-based, and fuzzy logic-based intrusion detection. Authors in [12] provided a survey based on virtual cloud security, discussing some IDS techniques for malware detection, and giving details on the related threat model and insights about the concept of cloud resiliency. Compared to this paper, previous works considered only CC, possible vulnerabilities and attacks in the CC environment, and related intrusion detection and prevention systems, but they did not deal with MCC.

Nevertheless, there are several survey articles in the field of MCC [13-22]. Among these survey papers, only [15-17] have discussed the security and privacy of MCC, comparing state-of-the-art works, analyzing different requirements, and presenting related open issues and challenges. Suo et al. [15] highlighted some privacy and security problems and existing approaches in MCC. Authors in [16] provided a survey considering the security challenges of MCC and investigated lightweight frameworks to enforce security and privacy in the MCC environment. In [17], the authors have proposed a more complete survey than those in

[15, 16] concerning privacy and security issues in MCC and how they raised due to the integration of mobile and cloud computing. On the other hand, the authors in [13, 18-20] provided a review of MCC, its challenges and perspectives without a specific focus on security and privacy. Sanaei et al. [14] also described MCC and its challenges, additionally discussing heterogeneity in this area. In [22], Wang et al. surveyed applications of MCC, its related challenges, and some available solutions; they also considered open issues and future directions of MCC applications. Authors in [21] focused on multimedia applications that leverage MCC and studied their technical issues and possible research directions.

### *1.1. Motivations and Contributions*

As demonstrated in the previous section, the use of CC and MCC is constantly growing today. Since the nature of this computing is to delegate the processing operations to the cloud environment to decrease users' workload and save energy and storage, privacy, and security concerns arise. In particular, security in CC and MCC scenarios has received wide attention from the scientific community. However, a comprehensive investigation of CI-based IDS techniques applied in CC and MCC environments is still lacking. Hence, filling this gap is the main motivation of our study.

In detail, this paper first attempts to provide insights about CC and MCC, then it studies and evaluates the proposed CI-based IDSs in both CC and MCC environments. We provide a critical review of the previous literature, intending to highlight the advantages and limitations of the state-of-the-art approaches proposed. Equally important, we aim to define a taxonomy for IDSs and we specifically classify CI-based techniques into single and hybrid methods. We refer to a single method as a single CI algorithm applied to the IDS for CC or MCC (e.g., one neural network leveraged to detect an attacker or a malicious behavior). On the other hand, a hybrid method is a combination of (at least) two CI techniques (e.g., a neural network combined with a fuzzy system used to identify or classify the attacker or malware). To the best of our knowledge, our survey provides this novel cross-section for the first time. Finally, we put together the considered aspects to point out open issues and future directions of research on this topic.

To emphasize the novelty of our in-depth investigation, Table 1 shows the comparison between this survey and the most-related ones previously discussed. It is worth noting that the *scope of our paper* is specifically focusing on technical articles that describe the function of *IDSs based on CI techniques* (e.g., fuzzy sets, NN, SVM, etc.) *in both CC and MCC environments*—providing also an overview of both paradigms—while other survey papers considered CC or MCC separately and took into account the CI-based IDSs *only for the CC environment*. Moreover, we explicitly bring out related *open issues* and give insights into the most interesting aspects considered in state-of-the-art literature (i.e. datasets, attacks, and possible countermeasures).

Given these considerations, the contributions of the present work are manifold:

- We present an overview of CC and MCC service models together with their applications, benefits, and possible shortcomings.
- We describe the IDS techniques applied in CC and MCC environments outlining their type, detection method, features, and limitations.
- We provide a comprehensive taxonomy of CI-based IDSs in both CC and MCC. To this end, we have searched and studied related papers that have been published since 2010 over Google Scholar, based on the occurrence of the IDS, CC, MCC, and CI keywords in the title and the content of the papers. Specifically, for each work, we identify the aim and the cloud environment it targets, together with its strengths, limitations, the dataset used for experimental evaluation, and the specific CI-based techniques employed.
- We further categorize these latter techniques into single and hybrid methods and further divide each group into subgroups based on the intelligent technique they use.

- We provide a discussion of most-related literature and compare the state-of-the-art methods through a common performance evaluation benchmark.
- Finally, we devise and organically present open issues and future perspectives of CI-based IDSs in CC and MCC.

The rest of the manuscript is organized as follows: in Section 2, we give a brief overview of CC and MCC models and IDSs. Section 3 consists of CI-based IDS techniques, their performance evaluation, and the advantages and disadvantages of single and hybrid methods. Section 4 and 5 present the open issues and conclusions of this paper, respectively.

Table 1: Comparison of this work with literature.

| ***Article*** | ***Overview of MCC*** | ***Overview of IDSs*** | ***Open Issues*** | ***CI-based IDS in CC/MCC*** |
|---|---|---|---|---|
| Patel et al. [4] | | √ | | |
| Modi et al. [3] | | √ | | |
| Modi et al. [7] | | √ | √ | |
| Iqbal et al. [8] | | √ | √ | |
| Osanaiye et al. [9] | | √ | √ | |
| Kumar et al. [11] | | √ | | Only for CC |
| Mishra et al. [10] | | √ | √ | Only for CC |
| Denz et al. [12] | | √ | | |
| Khan et al. [16] | √ | | | |
| Mollah et al. [17] | √ | | √ | |
| *This Paper* | √ | √ | √ | √ |

## 2. Brief Review of Cloud Computing, Mobile Cloud Computing, and Intrusion Detection Systems

In the following subsections, we present an overview of CC and MCC, discussing both basic definitions and concepts, and their applications. Then, we concentrate on IDSs with a specific focus on the IDS methods used in CC and MCC environments.

### *2.1. Cloud Computing*

As mentioned earlier, CC refers to a computation that is run by several remote servers, which are connected by a network and leads to centralized data storage and online access to computer resources and services. Based on the capabilities of CC [23], the following service deployment models are defined for cloud environments.

**Public cloud**: In this model, a service provider makes the resources available to the public through the Internet. The services offered via the public cloud may be free or pay-per-use based on resources employed [1]. In the public cloud, the service provider is responsible for the management and maintenance of hardware and application infrastructure. Therefore, it can save companies from high costs. Microsoft Azure services

platform [24], Amazon EC2 [25], Google AppEngine [26] and IBM's Blue Cloud [27] are some examples of public cloud services given by different providers.

Table 2: Pros and cons of cloud models [2].

| *Model* | *Pros* | *Cons* |
|---|---|---|
| **IaaS** | ✓ Pay-per-use<br>✓ Reduces total cost of ownership<br>✓ Elastic resources<br>✓ Better resource utilization<br>✓ Supports green IT | × Security issues<br>× Interoperability issues<br>× Performance issues |
| **PaaS** | ✓ Quick development and deployment<br>✓ Reduces total cost of ownership<br>✓ Supports agile software development<br>✓ Different teams can work together<br>✓ Ease of use<br>✓ Less maintenance overhead<br>✓ Produces scalable applications | × Vendor lock-in<br>× Security issues<br>× Less flexibility<br>× Depends on Internet connection |
| **SaaS** | ✓ No client-side installation<br>✓ Cost savings<br>✓ Less maintenance<br>✓ Ease of access<br>✓ Dynamic scaling<br>✓ Disaster recovery<br>✓ Multitenancy | × Security<br>× Connectivity requirements<br>× Loss of control |

**Private cloud**: Resources of a private cloud are dedicated to a single organization or client and they are not available to the public. Levels of security and control in this model are stricter than public clouds. The ability to customize the network components and storage is another advantage of a private cloud [1].

**Hybrid cloud**: The combination of public cloud and private cloud is defined hybrid cloud. This structure allows organizations to use public cloud services alongside their private cloud service management. Flexibility is one of the benefits of a hybrid cloud. In this case, the use of public cloud and private cloud resources can be changed when needed. For example, with workload increasing, processes are divided between public cloud and private cloud for better performance of the services [1].

**Community cloud**: Community cloud is used where several organizations have the same concerns and want to use CC by sharing the infrastructure. These organizations or a third party can manage and control the Community cloud [1].

According to the definition of NIST, CC consists of three main service models called SaaS, IaaS, and PaaS. **IaaS** service model provides access to main resources such as VMs, virtual storage, virtual infrastructure, etc. All these resources are available to the end-users through server virtualization, while the cloud service provider manages the infrastructure. Eucalyptus, Google Compute Engine, FlexiScale, Linode, and Amazon EC2 are some of the providers that offer IaaS service model. **PaaS** provides a runtime environment and tools for developing applications. It is a useful development environment in the cloud. In this model, clients are responsible for the services and applications that they develop, and the cloud service provider manages the rest of the operations. Red Hat OpenShift, Force.com, Google AppEngine, and

Windows Azure Platform are some popular PaaS providers. **SaaS** is a model that enables clients to use applications provided by the cloud over the Internet such as email and office tools. In this model, the service provider is responsible for controlling the network, infrastructure, storage, etc. and all application data and underlying infrastructure are situated (placed) in the data centers of the service provider. Microsoft Office 365, Google Apps, Oracle On-Demand, and Saleforce.com are some of the providers of this cloud service model. Table 2 summarizes the pros and cons of each of these methods [2].

According to published papers, there is a lack of standard architecture for the CC paradigm. For instance, in [16], the authors presented a layered architecture that includes the application layer, platform infrastructure layer, software infrastructure layer, supervisor software layer, and cloud's backbone layer for CC. Differently, C. N. Modi and K. Acha [7] considered a cloud architecture made of two ends, namely the front end and back end; the users leverage the former to access the cloud services, whereas the latter delivers these services. In [19], the authors considered a service-oriented architecture. Finally, a web-oriented architecture and a market-oriented architecture are presented in [28] and [29] respectively, where the CC architecture is defined based on the different services offered by the environment, as shown in Figure 1.

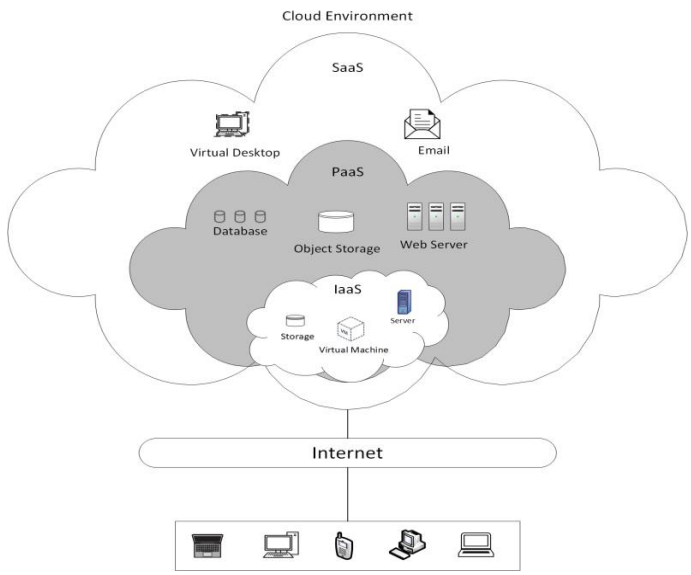



Figure 1: Overview of CC architecture [130].

### *2.2. Mobile Cloud Computing*

Recently, the use of smartphones and mobile devices is increasing. Cisco Visual Networking Index [132] has estimated that in 2022 Wi-Fi and mobile networks will account for 79% of Internet traffic as compared to 65% of 2017. These mobile devices use the mobile network to access the services, but they are battery operated so they suffer from limitations in storage, processing power, energy, communication, and security. These limitations lead to inefficient functionality of applications that need complex computations and storage. For this reason, it is better to offload these computations via wireless communication to an external computing device such as a cloud [30]. Accordingly, a new branch of computing named MCC has been formed.

Authors in [17], define MCC as the integration of wireless technology, CC and mobile computing, and mobile (or other) clients who can use several types of cloud services. Another definition of MCC according to [16] is described as a service that permits mobile clients with limited resources to adjust abilities such as storage and processing via offloading and dividing the jobs that need more storage and intense computations to cloud servers through wireless networks.

Figure 2 shows the MCC architecture. Firstly, mobile users are connected to the mobile network using a base station and they can use the mobile cloud services. Then, leveraging these services, the users can interact with the cloud via the Internet [16].

MCC includes different service models similar to CC [133, 150]. In the **mobile app as a service** model, clients use and run cloud mobile apps via the wireless network. In the **mobile network as a service** model, with the network infrastructure that is presented by service providers, clients can build their network and manage it. In the **mobile community as a service** model, clients can utilize services of the social community that is created and managed by a mobile clients' team. Another service model is the **mobile multimedia as a service** that allows clients to execute multimedia services via wireless media. In the **mobile data as a service** model, a group of services that depend on a database is presented for clients by providers to perform their data-based functions. Another model is the **mobile cloud IaaS** model; in this model, the service providers directly present the cloud storage and infrastructure to the users [17].

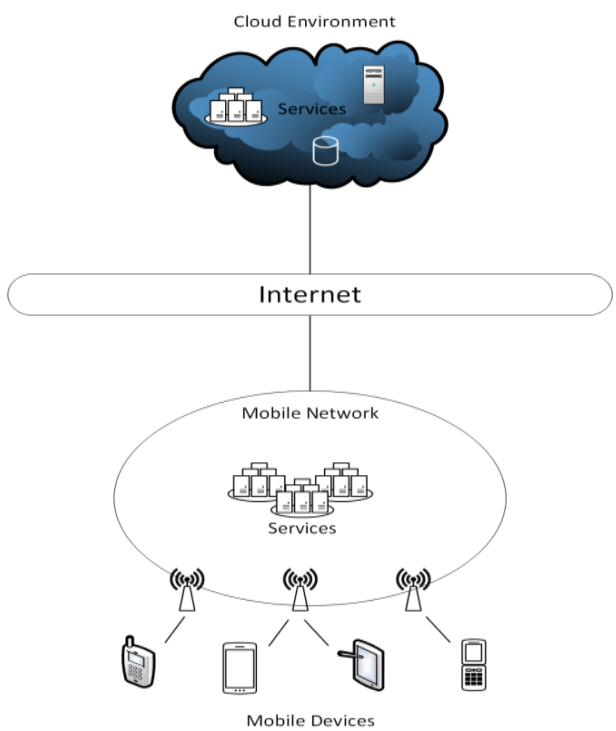



Figure 2: Overview of MCC architecture.

### 2.2.1. MCC Applications

As the utilization of mobile networks and mobile devices grows, this leads to an increase in network capacity, and mobile devices reach considerable computational capabilities, making new applications possible in mobile scenarios. The applications of MCC are as follows: mobile gaming, mobile healthcare, mobile commerce, mobile learning, etc. In mobile gaming, all of complex works and computations are sent to the cloud to preserve a mobile device's resources such as battery and memory [31, 32]. Mobile commerce is another application of MCC that is used in shopping, finance, and advertising [33, 34]. The aim of using MCC in healthcare applications is to reduce the limitations of traditional methods such as security and physical storage problems in medical treatment. This application allows mobile users to access their records easily from anywhere [35-37]. In mobile learning applications, mobile devices are utilized for training and increasing the users' ability. This is obtained thanks to the combination of mobility and e-learning technologies [38-40].

### 2.2.2. MCC Benefits

In MCC, program execution and data storage take place on a cloud server. Thus, the reliability will improve, facilitating the preservation of information and avoiding data loss. Also, CPU processing rates and storage capacities of mobile devices are limited. Changing the hardware of a mobile device may be one of the solutions but it causes additional costs. Mobile users can access their data files that have been already stored on the cloud server or delegate demanding computation to the cloud infrastructure by using wireless technology that MCC provides. Offloading is also a way provided by MCC to decrease the energy consumption of battery-powered mobile devices [41-43]. Lastly, we underline that since MCC is a mixture of computing and mobile computing, the advantages of CC such as multitenancy, ease of integration, scalability, and dynamic provisioning are also extended to this scenario [192]. Figure 3 depicts a diagram that summarizes both the MCC applications and service models previously described.

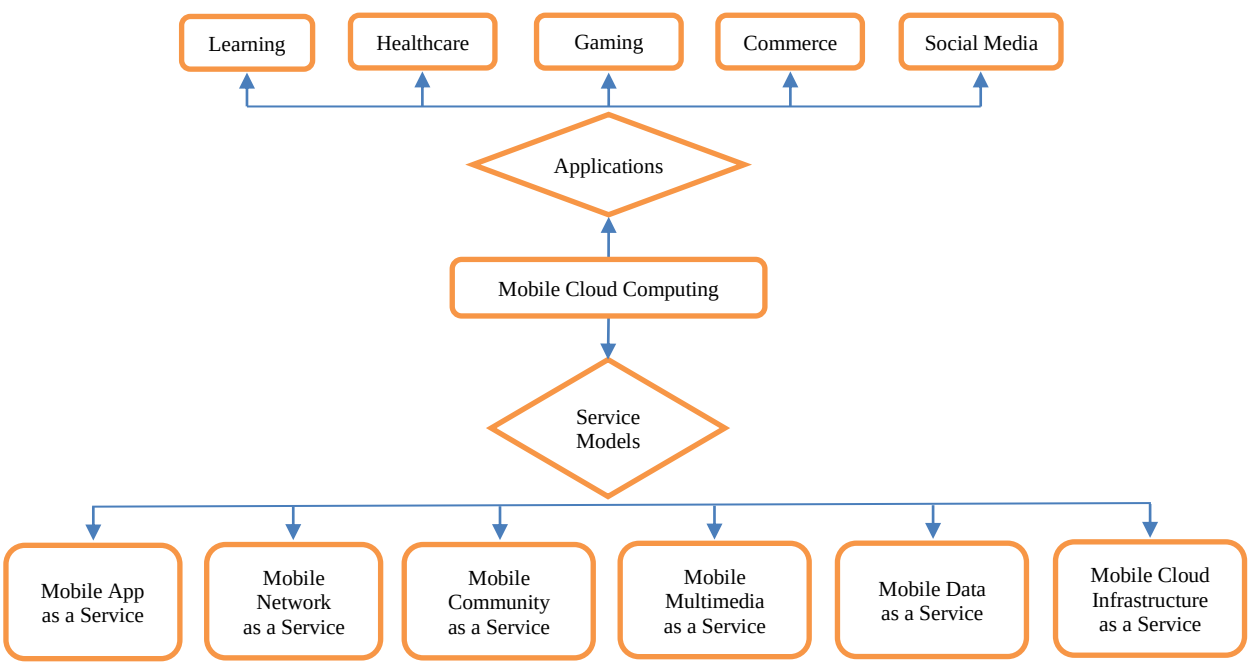



Figure 3: Mobile Cloud Computing Summary Diagram.

### *2.3. Intrusion Detection Systems*

In general, two kinds of threats can be considered in CC and MCC: an insider threat and an outsider threat. An insider threat is a kind of threat that comes from users within the company or organization such as employees or ex-employees who have access to computer systems and data. An outsider threat refers to any kind of threat that comes from outside entities. These threats can affect the integrity, availability, and confidentiality of cloud resources. Using firewalls is a solution to this kind of security attacks.

A firewall can protect a network or a computer system from unwanted and unauthorized traffic. There exist different types of firewalls [44, 45]. A basic type of firewall is known as stateless inspection firewall or packet filtering firewall. These firewalls typically work at the network layer and investigate the header's information of packets such as protocol type and source and destination IP addresses according to some defined rules. Therefore, packet filtering firewalls are not generally suitable for fragment and spoofing attacks. Another type is the stateful inspection firewall that maintains the connection state in a state table. In this case, the packet's header information is controlled by the firewall. Then, the firewall compares this information with the state table to check if they match or not. Application firewalls can monitor and control inputs and outputs of any service or application and operate at the application layer. All traffic from a lower layer to the application layer is controlled by this type of firewall. Sometimes proxy agents are used for communication across two hosts. This attribute of a firewall is named application layer gateway or proxy. Unlike application firewalls, the proxy firewalls have superior security level and decryption capability to inspect coded payloads. Another type is the firewall dedicated to virtualized infrastructures. In this model, firewalls operate on VMs and control the connection among them and perform packet filtering. Table 3 reports a brief overview of the aforementioned types of firewalls.

Table 3: Types of Firewalls.

| **Firewalls Technology** | *Firewalls for Virtual Infrastructure* | *Proxy Firewalls* | *Application Firewalls* | *Stateful Inspection Firewalls* | *Packet Filtering Firewalls* |
|---|---|---|---|---|---|
| **Description** | • Dedicated to virtual infrastructures<br>• Can add extra cost to configure VMs | • Also known as application layer gateway firewall<br>• Need more resource<br>• Examine the actual content of traffic | • Control traffic from low layer to application layer<br>• Can recognize a series of unwanted commands | • Improvement of stateful packet filtering firewall<br>• Maintain the state table of the open connections<br>• Operate at the IP level<br>• Can detect IP spoofing and DoS attacks | • Cannot detect malicious codes<br>• Examine only the header of the packets<br>• Not suitable for fragment and spoofing attacks |

Firewalls generally can detect outsider attacks, but they are not capable to detect insider attacks. Therefore, another solution for the detection and prevention of network attacks is to use IDS and IPS systems. To recognize security breaches, an IDS collects and analyzes network or computer system information [46].

Previous works have presented various classifications of different types of IDSs. One of these classifications is based on data sources [47]. This means that an IDS monitors different data sources or environments to detect malicious attacks. NIDS, WIDS, HIDS, AIDS, and NBA are specific types of this categorization according to [47-50]. NIDSs and WIDSs monitor, collect and analyze network and

(specifically) wireless network packets, respectively, to detect attacks. The HIDS collects data from servers or host computers to verify the intrusion. It can detect attacks that are not verified or observed by NIDS. AIDS is a subcategory of HIDS which specifically supervises application data and log files for suspicious activity. An NBA system also monitors the network traffic enforcing signature- and behavior-based detection to identify unknown threats and doubtful behavior of the network.

Signature-based and anomaly-based detection are two major methods to enforce intrusion detection. The signature-based detection method performs a comparison between captured events and attack patterns to identify the feasible intrusions [48]. In the anomaly-based IDS, a baseline profile of a normal network or system activity is created, then each incoming packet or event that deviates from the defined baseline is taken as an intrusion [46]. Each of these methods has its benefits and drawbacks. Signature-based detection systems can detect known attacks with low false-positive rate but keeping patterns/signatures up to date is a major drawback of this method. Anomaly-based detection systems are effective for detecting unknown or new attacks, however high false error rates and system's inner complexity are examples of the drawbacks of these systems [46]. In Table 4 we provide an overview of the different types of IDSs (cf. Section 2.4 for details on HVIDS) with their pros and cons.

Table 4: Types of IDSs.

| **Type of IDS** | *NIDS* | *HIDS* | *NBA* | *WIDS* | *DIDS* | *HVIDS* |
|---|---|---|---|---|---|---|
| **Location** | In the virtual network or external network | On the host system, virtual machine, and hypervisor | In the internal or external network | In the internal network | On the VMs, hypervisor, host or external network | On the hypervisor |
| **Features** | • Can analyze the activity of protocols and applications<br>• Suited for offending attacks | • Controls individual host<br>• Can analyze the activity of encrypted connection | • Examines network traffic<br>• Discovers attacks with unanticipated traffic stream<br>• Suited for zero-day and new malware attacks | • Can monitor wireless traffics and protocols<br>• Has high accuracy | • Combines HIDS and NIDS models<br>• Has features of both HIDS and NIDS | • Can capture the state information of each VM |
| **Constraints** | • Can't detect encrypted traffics<br>• Not suitable for host systems | • Unable to find attack records<br>• Unable to detect evasion attacks | • A delay can occur in some cases<br>• The number of false positives increased in some situations | • Unable to control the activity of network, transport and application layers<br>• Vulnerable to jamming attack | • High cost in terms of communicat ion | • May have a high false-positive rate |

### *2.4. IDS Methods Used in (M)CC*

As explained in the previous section, IDS systems can be grouped into different types and use various techniques to identify attacks and threats. In this section, we discuss various types and techniques of IDS that are utilized in the CC environment and, given their nature and technology employed, seamlessly integrated in the MCC one. Figure 4 shows the categorization of IDSs based on different viewpoints. Generally, the five perspectives taken into account are the detection method, structure of IDS, detection time, IDS type, and response time. For what concerns this survey (i.e. the IDSs leveraged in CC and MCC environments) we will focus on two specific aspects, namely the detection method and IDS type.

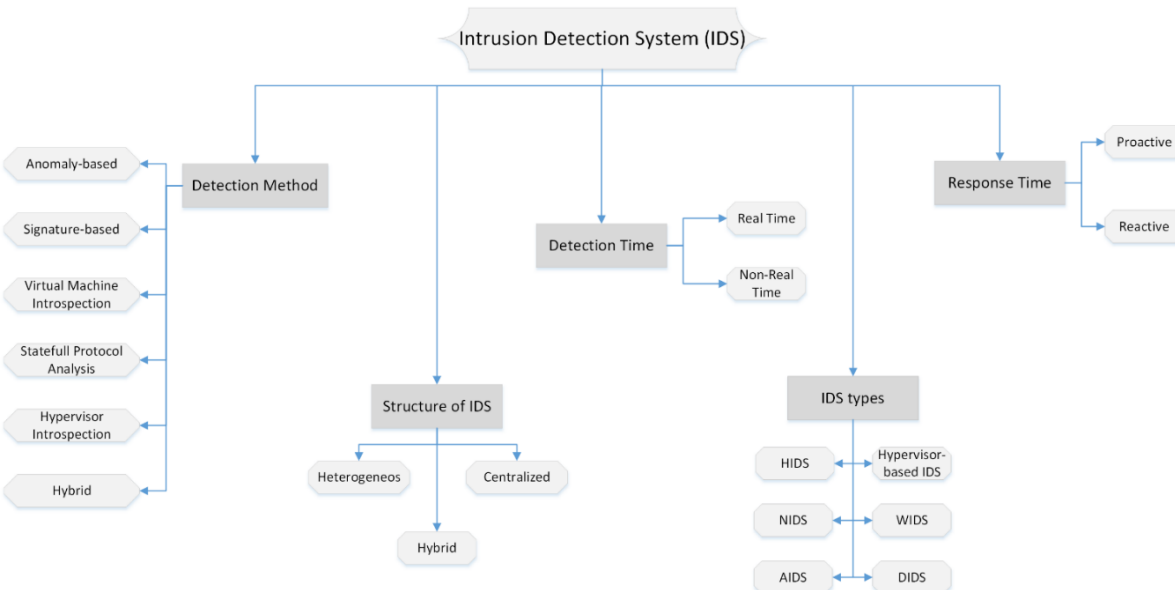



Figure 4: Classification of IDSs.

Besides HIDS and NIDS that are used in general IDSs, HVIDS and DIDSs are other types of IDSs employed in (M)CC which are discussed in [3, 7] and depicted in Figure 5. DIDS is made of several NIDS and/or HIDS, thus it profits from both systems advantages [7]. Hypervisor, also known as virtual machine monitor or VMM, is the computer software that can control VMs and can host several operating systems simultaneously [64]. An HVIDS can control the traffic flows between VMs or the VM and the hypervisor [7]. Nikolai and Wang [61] proposed an HVIDS for clouds. Their method performs control operation from outside of the VM and demonstrates that HVIDS is a good choice for the detection of DoS attacks both from and against a cloud model.

IDS techniques used in CC are generally divided into three groups: anomaly-based, signature-based, and hybrid intrusion detection. Definitions of anomaly-based and signature-based intrusion detection techniques have been discussed earlier. Mishra et al. [10] added two more groups for IDS methods in a cloud that includes Virtual Machine Introspection or VMI and Hypervisor Introspection or HVI (cf. Figure 6).

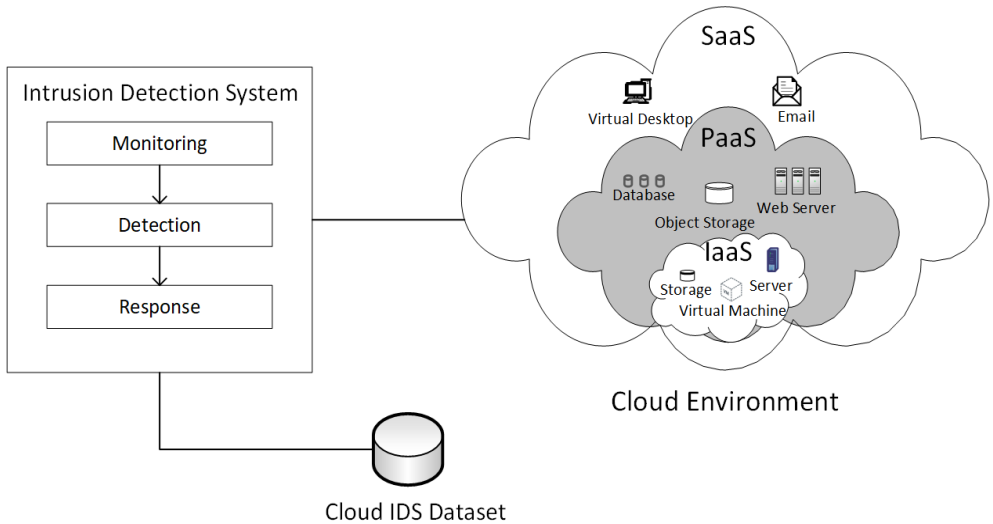



Figure 5: Example architecture of IDS in CC [136].

VMI is a collection of techniques used to control or monitor the condition of guest OSs, VMs and application software and it is performed at the hypervisor layer [65-67]. The VMI IDS is designed to discover insider attacks from one VM against another or from a VM against the VMM [10]. In CC, VMI is useful for the VMM to monitor the behavior of a VM. Specifically, the VMI IDS can analyze the state and events of a VM. Detection systems that are located on a host can provide a good level of visibility, but they are vulnerable to attacks. By installing the IDS on the network and out of the host, its vulnerability is decreased, and it is more resistant to attacks than the previous case, but it has a low level of visibility. The VMI IDS utilizes the information of events and hardware states that are observed directly to extract the host's software state. VMI inherits visibility and isolation of both HIDS and NIDS respectively, therefore, it represents a good choice for host monitoring from the outside [68]. Authors in [66], introduced an IDS method based on VMI for anomaly detection in VM, named Collabra. It is combined with each hosts' VMM and examines the correctness of hypercalls. The authors considered anomaly detection since it is more suitable for detecting hypercalls. In [69], Mishra et al. introduced a VMI-based malware attack detection deployed at the VMM. This method utilizes a Markov chain to generate a system calls dependency graph. By doing this and utilizing an injection method, authors extracted the traces of a system call of malware to make the system protected against evasion attempts.

HVI is used to provide security for VMs that are running at the hypervisor level of the cloud environment. These types of techniques can discover a hardware attack, a hypervisor attack, and a VM-host OS attack [10, 70]. The main part of VMI IDS that enables a safe place for executing the tools of VMI is HVI. Indeed, in HVI, control flow data, hypercalls and data structures are examined for the protection of hypervisors [10]. Authors in [70] introduced an HVI method to discover hypercall injections. This method can extract the hypercalls to control the performance of guest VMs from outside the hypervisor.

For the sake of completeness, Table 5 summarizes IDS techniques employed in CC and MCC environments characterizing them according to their detection method and type and emphasizing also their peculiar features and limitations.

Table 5: Existing IDS techniques in CC and MCC.

| ***Article*** | ***Detection Method*** | ***Type*** | ***Feature(s)*** | ***Limitation(s)*** |
|---|---|---|---|---|
| Derfouf et al. [51] | Signature Detection | HIDS | • Controls VMs and hypervisor<br>• Beneficial in terms of cost and portability | • Unable to discover unknown attacks |
| Deshpande et al. [52] | Anomaly Detection | HIDS | • Examines system calls<br>• Reduces computational load<br>• Increases accuracy rate | • May have a delay in attack detection to provide better accuracy rate |
| Wang & Zhu [53] | Anomaly Detection | HIDS | • Consumes fewer resources like memory and CPU | • Spends more time than competitors |
| Mahajan & Peddoju [54] | Signature Detection | DIDS | • Discovers attacks on VMs | • Not suitable for unknown attack detection |
| Salek & Madani [55] | Signature Detection | NIDS | • Improves time and resource consumption<br>• Reduces drop rate of packets | • Switching among trust levels is not dynamic |
| Balamurugan & Saravanan [56] | Signature Detection | Cloudlet controller | • Decreases rate of packet loss<br>• Increases throughput and speedup ratio | • Not tested for real-time environment |
| Ram [57] | Signature Detection | DIDS | • Able to discover DDoS attacks | • More computationally demanding than pure Snort-based IDS |
| Velliangiri & Premalatha [58] | Signature Detection | NIDS | • Increase the accuracy rate<br>• Discovers DDoS attacks | • Designed to discover only (D)DoS<br>• Computational complexity not taken into account |
| Sandar & Shenai [59] | Signature Detection | NIDS | • Can detect EDoS attack | • Proof-of-concept able to discover only EDoS |
| Ghorbani & Hashemi [60] | Signature Detection | DIDS | • Decreases time of processing<br>• Scalable approach | • Scalability and false alarms not checked |
| Nikolai & Wang [61] | Signature Detection | HVIDS | • Discovers DoS attacks<br>• Controls VMs from outside | • High false alarm rate |
| Lombardi & Di Pietro [62] | Virtual Introspection | HVIDS | • Can detect most of the attacks | • A little penalty in performance<br>• High cost |
| Ghorbani & Shahrezaie [63] | Signature detection | DIDS | • Can enhance the efficiency of IDS<br>• Reduces the time of processing | • May have high cost for attaching global features |

## 3. Computational Intelligence-Based IDSs for CC and MCC

In this section, we study various types of CI techniques that have been used for IDSs in CC and MCC. Based on the IDS types and detection methods discussed in Section 2, Figure 6 shows the classification of IDS methods in CC and MCC delving into the specific techniques employed by each method, being the various IDS types cross-cutting to this classification. In the following subsections, we start reviewing well-known attacks faced by CI-based IDSs in CC and MCC environments and the datasets leveraged by researchers in the most-related literature on this topic. Then, we further cluster CI techniques for IDSs into single methods and hybrid methods based on the number of intelligence techniques used in each paper and present a brief description of each method. Finally, we analyze cooperative methods that improve the detection accuracy using multiple IDSs and enter into details of the CI-based IDS solutions designed for MCC.

### *3.1. Attacks in CC and MCC*

In the following, we provide a brief review of some known types of attacks in cloud and mobile cloud environments. Network attacks refer to any method that can compromise the security of the network by exploiting its vulnerabilities. These attacks can interrupt the network functionality, decrease the throughput, deplete bandwidth and network resources, deny services, etc. Data modification, IP spoofing, eavesdropping, brute force, password cracking, man-in-the-middle, and denial of service (DoS) attacks are common types of attacks afflicting cloud networks [131, 155, 190].

DoS is a kind of attack that floods a target system with many requests to use its resources in the extreme and makes the service unavailable for legitimate users. In a DDoS attack, a group of infected computer systems, which is known as zombies, floods with malicious packets the target server to slow down its services or make the services unavailable for its users. DDoS attacks are categorized into two types, reflector, and direct attack. In reflector attacks, an attacker forwards requests to reflector hosts (potentially legitimate servers) by spoofing an IP address set to that of the victim, then each host forwards the responses to the victim server. In a direct attack, the attacker uses zombie computers for directly sending malicious packets to the target server. HTTP, XML, ICMP, TCP, and UDP flooding attacks are examples of direct attacks, while the Smurf attack, based on ICMP Echo request packets, is an example of a reflector attack [114].

In a user-to-root (U2R) attack, the attacker sniffs the password of a user and gains access to its account. After this by exploiting known bugs or backdoor in the target system, the attacker can gain root privileges and then can compromise other systems by performing malicious tasks. A known example of this type of attack is the buffer overflow which occurs when data are added to a buffer to exceed its size and overwrite adjacent memory locations (e.g., to inject malicious code).

The remote-to-local (R2L) attack is a kind of attack which considers unauthorized local access from a remote machine. An attacker gets access to a machine as a valid user by utilizing the existing weaknesses of this machine's security.

In a network probe attacks, the attacker scans the network to collect information and find vulnerabilities (e.g., open ports) to launch an attack [115, 116]. Port sweep attack and IP sweep attack are two kinds of probe attacks that scan the host or the whole network to find out open ports and IP addresses, respectively. Nmap, which is a network monitoring tool, can be used to generate network probe attacks.

Advanced Persistent Threats leverage several advanced methods (e.g., injection of different malware types) to repeatedly steal data and other sensitive information. Notably, the common targets of these attacks are cloud storage systems. These attackers can also induce the storage system to apply a particular defensive strategy then attack with the aim of defeating the induced defense.

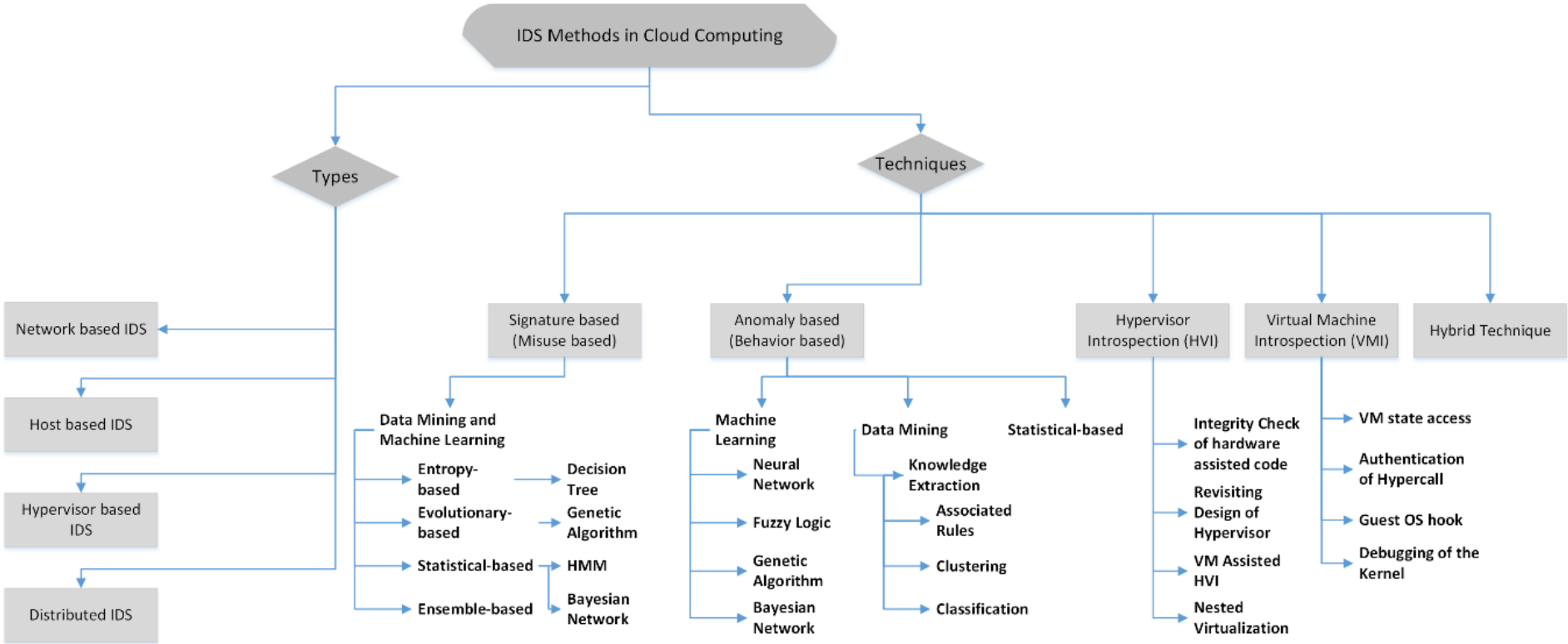



Figure 6: Classification of IDS methods employed in CC and MCC [10].

In addition to traditional attacks, there exist other types of attacks or threats against virtual environments that are discussed next. A side-channel attack is a type of attack in which the information of side channels like power, cache, and time is exploited to get access to sensitive information of a VM. In detail, the attacker employs side channels to bypass the existing isolation among VMs [117, 181].

A kind of attack in which an intruder achieves the ability to read and write his content by accessing another host operating system or VMs' memory is known as VM escape. This can be a severe attack for VMs, particularly in the (M)CC environment [182].

Hyperjacking is another kind of attack in virtual environments. In this attack, the entire control of a server is taken by placing a rogue hypervisor so that the attacked VM is completely oblivious to its presence. For instance, one reason for hyperjacking can be to install rootkits. Hyperjacking is done when the attacker gains direct access to the main hypervisor [118].

Hypercall vulnerabilities are another attack that occurs when a malicious guest penetrates the VM with the utilization of hypercall interfaces (used by the guest OS to make privileged requests) and gaps in the hypercall handler of the hypervisor. This attack causes substantial performance degradation of the hypervisor [119].

The infrastructure of the MCC has a distributed and open nature, therefore mobile devices and cloud resources are attractive for attackers to violate the security. Because of the similarities between a computer system and a mobile device, existing attacks in computer systems can also be seen in mobile devices such as trojan, rootkits, botnet, worm, and virus [171]. Also, the cloud attacks can be spread to mobile devices via resource sharing among them [120].

Based on the studies in security mechanisms for CC and MCC, using intrusion detection techniques is a better solution than cryptography and firewalls for both CC and MCC environments. Additionally, for MCC it is better to place an IDS in the cloud environment because of the limited resources of mobile devices [121]. Table 6 shows the type of attacks, their possible consequences, and papers from the literature that have proposed countermeasures applied to mitigate the specific attacks. We will give the details of CI-based countermeasures enforced by each referenced work in the next subsections.

Table 6: Type of attacks, mechanics, and countermeasures.

| ***Attack Type*** | ***Consequence*** | ***Countermeasures*** |
|---|---|---|
| DoS & DDoS | • Affect service availability | [71, 73, 74, 84, 87, 99, 100, 122] |
| XML Injection | • Insert malicious content into message<br>• Perform unauthorized action<br>• Access to sensitive information | [71, 92, 122] |
| SQL Injection | • Repudiation issues such as voiding transactions or changing balances<br>• Disclosure of data on the system or making it unavailable | [71, 180] |
| Network Probe Attack | • Scan the network to obtain vulnerability information | [87, 99, 100, 101] |
| User to Root | • Can gain root level access to a VM | [87, 99, 100] |
| Remote to Local | • Compromise the installed hypervisor to reach control over the host | [100, 101] |

| | | |
|---|---|---|
| Advanced Persistent Threats | • Steal sensitive data from a certain target by exploiting vulnerabilities using various attack techniques | [177, 178] |
| Side-channel Attack | • Access to sensitive information of a VM via side-channel backdoors | [117, 181] |
| VM Escape | • Access (r/w) victim data through another host operating system or VM memory<br>• Particularly severe in MCC | [182, 121] |
| Hyperjacking | • Take control of a server by employing a rogue hidden hypervisor<br>• Allow to install rootkits and gain direct access to the hypervisor | [118, 121] |
| Hypercall Vulnerabilities | • Cause substantial performance degradation of the attacked VM | [119, 120] |

### *3.2. Datasets*

According to state-of-the-art literature, most of the researchers performed experimental validation and evaluation of proposed approaches through various public datasets, while others created their own datasets for CI-based IDS over CC and MCC. Hereafter, we will give a brief description of most-commonly used public datasets, whose summary is also shown in Table 7.

The KDD Cup 1999 dataset [126] has been utilized for assessing anomaly detection techniques since 1999. It is derived from the DARPA 1998 dataset that contains military network intrusions used by researchers to develop Machine Learning-based classification and clustering algorithms with a specific aim on security. In the KDD Cup 1999 dataset, the number of training data is about five million connection records spanned over three weeks. The training data are labeled as normal traffic or with the attack type and consist of categorical and statistical features. The dataset contains four types of attack DoS, User-to-Remote, probing, and Remote-to-Local, and it is usually used for network-based anomaly detection systems [123]. The KDD Cup 1999 dataset consists of three different classes of features including traffic features, content features, and basic features. Features related to TCP connections are gathered in the basic class. Features related to a window interval are grouped in traffic class and finally, the content features are invoked by the group of features that checks the behavior of the data section that is suspicious.

Another widely used dataset is the NSL-KDD [127] that is an enhanced version of the KDD Cup 1999 introduced to overcome the existing issues of this latter. The size of the NSL-KDD dataset is smaller than the KDD Cup 1999 because it eliminates the unessential and repeated records from this dataset. With this data reduction in NSL-KDD, randomness selection in the KDD dataset is not essential and experiments could be run on the entire dataset [124]. The features and attacks' types are the same as in the KDD Cup 1999 dataset, being its branch.

CIDD is a cloud intrusion detection dataset that contains behavior-based and knowledge-based audit data. It consists of real samples of both network- and host-based attacks. It helps to build intrusion detection techniques to discover various types of attacks such as User-to-Remote, DoS, Remote-to-User, probing, and masquerades among many others. To analyze the audit data from log files, its authors developed a Log

Analyzer and Controller System (LACS). In detail, LACS parses and analyzes user log files and correlates audit data based on IP addresses to generate the final statistical tables. Therefore, each table built by a LACS summarizes the behavior of a user [111].

More recent works leveraged the CIDDS-001 dataset [152], a labeled flow-based dataset created in a virtual environment using OpenStack for evaluating anomaly-based IDSs. Specifically, to generate the CIDDS-001 dataset, its authors emulated a small business environment including common clients, and email and web servers. They generated normal traffic according to the working schedule that differentiates between working hours and lunch breaks, and malicious traffic executing DoS, Brute Force attacks, and Port Scans within the network. Timestamps, origins, and targets of enforced attacks are used to label the recorded NetFlow data. Overall, the dataset comprises 24 time-based flow features related to 7 different network attacks.

CAIDA usually gathers various kinds of data and makes them accessible by researchers [129]. Some of its datasets are for special attacks or events. A list of different datasets together with category, source of data collection, status, availability, and release date are available and downloadable for scholars and researchers. Border mapping dataset, DDoS attack 2007, code red worms, witty worms, and many others are some datasets that are available by CAIDA [125].

The authors of the Uspscims project [160] aim to protect VMs against DoS attacks in the CC environment. They perform attacks from multiple VMs against another VM in a multi-tenancy CC deployed on the Eucalyptus platform. Moreover, they propose and implement an IDS that encompasses a packet sniffer, a feature extractor, and a classifier. The outcome of this experimental setup is a dataset of 5274 instances (4592 legitimate and 682 attacks), each containing 24 time-based traffic features. These are fed to the classifier component to discriminate between the attack and legitimate traffic. In detail, the proposed IDS is tested against TCP SYN Flood, TCP LAND, UDP Flood, DNS Flood, ICMP Flood, and Ping-of-Death (i.e. various types of DoS) attacks, showing high classification accuracy.

The ISOT-CID cloud dataset is introduced in [136] to overcome the lack of data coming from a real environment. To this goal, a production OpenStack cloud environment is leveraged to collect data from all the three cloud layers (cf. Sec. 2.1). Collected data include network traffic as well as logs and other information extracted from VMs and hypervisors. Several kinds of attacks are considered and categorized into inside and outside attacks, being conducted by an attacker internal or external to the CC environment, respectively.

The UNSW-NB15 [179] dataset is created to renew NIDS benchmarking that, as shown above, is still based mostly on the outdated KDD Cup 1999 and NSL-KDD datasets. Specifically, UNSW-NB15 is designed to overcome the limitations of previous, dated works and uses a hybrid configuration with real and synthetic traffic. A total of 49 features from traffic traces are extracted at flow granularity and nine different attack families in addition to benign traffic are considered.

The CICIDS2017 dataset [183] contains benign traffic and up-to-date common attacks and is provided both as raw data and traffic features. These latter are extracted via the CICFlowMeter tools for each labeled (bidirectional) flow defined based on the timestamp, source, and destination IPs, source and destination ports, protocols, and attack. The authors have built the abstract behavior of 25 users based on HTTP, HTTPS, FTP, SSH, and email protocols. Data are collected for a period of five days (3-7 July 2017). The implemented attacks include Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS.

The CSE-CIC-IDS2018 dataset [183] is an advancement of CICIDS2017 having the aim of defining a systematic approach for generating comprehensive benchmark datasets for IDSs on the basis of different user profiles, abstracting the representation of events and behaviors in the network. The combination of these profiles is employed to generate a set of different features. Specifically, the final CSE-CIC-IDS2018 dataset encompasses seven attack types: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of

the network from inside. Even this dataset is provided as raw network traffic captures and system logs of each machine (50 attacking machines and 420/30 victim machines/servers), along with 80 features extracted from the captured traffic using CICFlowMeter.

This overview highlights that only a very limited number of works in literature have collected and publicly released datasets in the context of CC/MCC security so far. Indeed, several studies in the field of CC and MCC security [87, 96, 99] leverage, for the evaluation of proposed solutions, public datasets that were not originally designed for the CC/MCC scenario, as in the case of KDD [126] and NSL-KDD [127]. Unfortunately, this setup impairs the applicability of the results obtained. On the other hand, a number of other cloud-specific datasets (e.g., Inside Dropbox [159] and Amazon S3 [186]), despite being valuable for the traffic characterization of CC/MCC environments, do not explicitly take into account the network security perspective and thus are out of the scope of this survey.

Table 7 recaps the datasets used for CI approaches in IDS over CC/MCC.

Table 7: Datasets used for CI approaches in IDS over CC/MCC.

| ***Dataset*** | ***Description*** |
|---|---|
| DARPA KDD CUP 1999 [126] | Traffic, content, and basic features of four types of attack (i.e. DoS, U2R, Probing, R2L) and one normal category |
| NSL-KDD [127] | An enhanced version of KDD CUP 99 with fewer records than KDD CUP 99 |
| CIDD [111, 128] | Misuse- and anomaly-based audit data encompassing real samples of both network- and host-based attacks |
| CIDDS-001 [152] | Emulated NetFlow data labeled with three types of attack (i.e. DoS, Brute Force, and Port Scans) |
| CAIDA [129] | Data collected from different sources and usually utilized for research purposes |
| Uspscims [160] | Dataset for detecting and preventing DoS attacks on VMs consisting of 24 time-based traffic features related to six DoS attacks |
| ISOT-CID [136] | Network intrusion detection dataset collected in a production OpenStack cloud environment, containing data from inside and outside attacks |
| UNSW-NB15 [179] | Network intrusion detection dataset comprising 49 TCP flow-level features from nine different attack families |
| CICIDS2017 [183] | Raw traffic captures and flow-level features related to benign traffic and up-to-date common attacks |
| CSE-CIC-IDS2018 [183] | Advancement of CICIDS2017 leveraging various user profiles to define different set of features |

### 3.3. Single Methods in CC

Hereinafter, we discuss the single methods leveraged in the reviewed literature. Table 8 summarizes in a practical way the CI-based single methods applied in cloud IDSs by these latter works, highlighting the particular technique utilized (i.e. Fuzzy Logic, Decision Tree, Genetic Algorithm, Game Theory, Support Vector Machine, etc.), their aim, pros and cons, the dataset and the cloud environment leveraged, and briefly their main contribution. Furthermore, to highlight the progress in the state-of-the-art, we have ordered by year the works reported in Table 8.

Chan et al. [71] proposed FAR and FAP intrusion detection and prevention systems to operate against web service attacks, specifically against SaaS CC. They used 20 fuzzy association rules and 336 fuzzy associative patterns for their IDSs/IPSs deployed over a public cloud platform. For the performance evaluation of these systems in terms of transaction time, they defined five operational scenarios. Results showed that the FAR-based IDS/IDP performs better than FAP-based IDS/IPS. These two fuzzy-based systems can detect and prevent known web (service) attacks (e.g., XML-DoS, XML injection, SOAP oversized, SQL injection, etc.) with less than 1% of false alarm rate and a high accuracy close to 100% detection rate.

Wang et al. [72] presented a behavior-based botnet detection technique that uses a fuzzy pattern recognition method. The proposed method included five stages, with the last two stages comprising a detection stage for DNS and one for TCP. The authors enhanced the detection accuracy by accurately adjusting the membership functions used by the fuzzy pattern. Results are evaluated via the true positive, false negative, and false-positive rates and compared with static analysis and data-mining approaches [184] feature stream [71], and fuzzy pattern [72]. As shown in Figure 7, the proposed method achieves less than 3.5% false-positive alarm rate and can detect about 95% of bots, outperforming all the considered baselines.

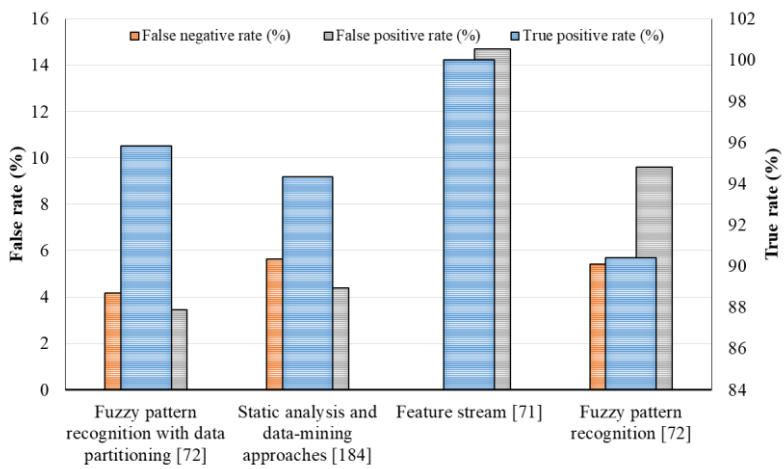



Figure 7. Comparison of behavior-based botnet detection techniques [72] with related baselines.

Watson et al. [73] presented an online anomaly-based detection method that utilizes the SVM algorithm to discover malware in the hypervisor level of the cloud. In this work, the authors used network-level and end-system data to extract the features and build the feature set. Their method performs well with system-based features and can detect anomalies with 90% detection accuracy, but with network-level features, the results are less accurate. However, their results showed that the proposed implementation of the SVM method for malware detection can detect anomalies with minimum time cost.

Iyengar et al. [74] introduced a protection method against DDoS attacks by using fuzzy logic. First, the authors surveyed some types of DDoS attacks and solutions to protect the cloud environment from these attacks. Then, they devised an IDS, installing a fuzzy system in the cloud which checks the input traffic to discover DDoS attacks. This system consists of four working stages. In the first stage, the designed rules of the fuzzy system are used to make a choice and specify the traffic type. In the second stage, the traffic is analyzed, and its type is assessed by this fuzzy-based IDS. In the third stage, the system triggers an alarm by discovering an anomaly Finally, in the last stage, a request is sent to the routers for malicious packet entry rejection. The described method can provide the availability and protection of cloud resources, decreasing the cost of data transmission and storage functionality.

Wang et al. [75] presented a signature-based and anomaly-based IDS for the detection of mobile malware. They employed the signature-based detection for known attack detection, whereas the anomaly-based one for zero-day and unknown attacks detection with the help of a linear SVM classifier (i.e. equipped with a linear kernel). Firstly, the SVM-based anomaly detection method checks the new application for being abnormal or normal; if this latter is abnormal, then in the second step a signature detection method distinguishes its type or class. Results were evaluated using the classification rate (viz. accuracy) for each malware family. Figure 8 presents the average classification rate over all the malware families of the proposed linear SVM and compares this latter with two simple variants of the Support Vector Classifier (SVC) subject to L1 and L2 regularization, respectively. Based on the results of Figure 8, the proposed method provides the highest average classification rate. Specifically, the proposed SVM can accurately classify malware and leads to low false-negative (1.16%) and high true-positive (98.94%) rates. However, it is inefficient for the detection of applications incorporating HTML5 or native codes.

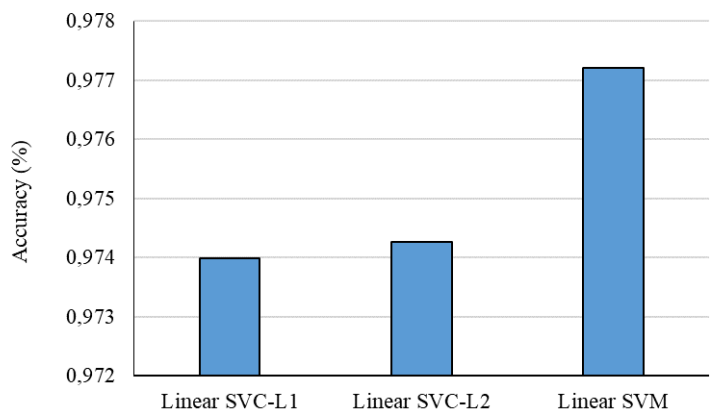
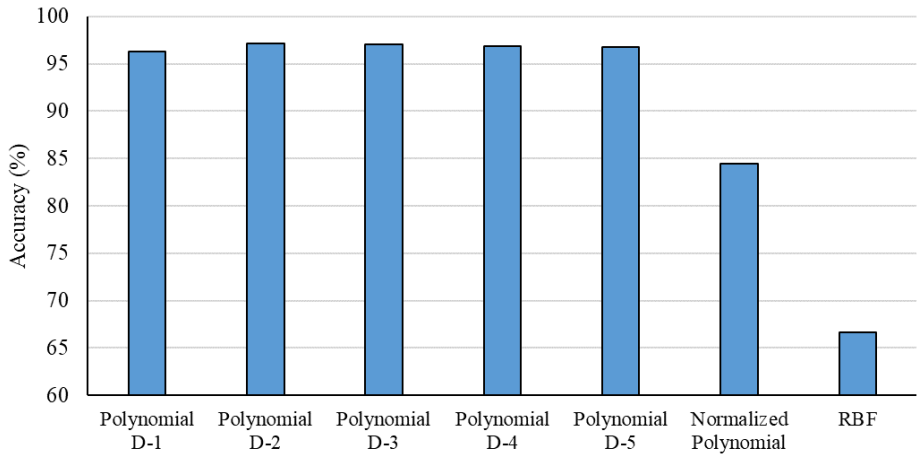


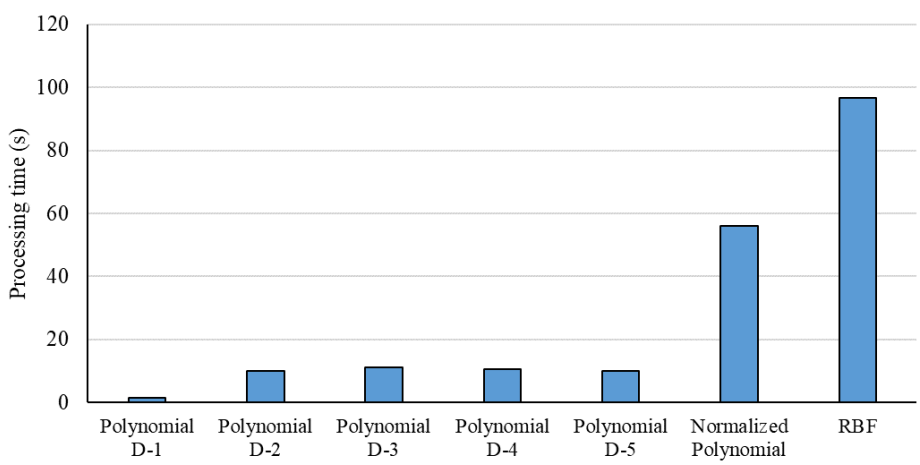

Figure 8. Average classification rate of the Linear SVM proposed in [75] compared with naïve SVC variants.

Khorshed et al. [76] presented another paper in the field of CC security. They studied several security issues in CC and provided a proactive approach to discover the threats. It determines its information according to the threat pattern of attack detection and notifies the administrator or the user about the threat. Authors utilized an SVM technique for the detection of cloud attacks and compared it with other machine learning techniques including PART [77], DT [78], multilayer perceptron [79], and Naïve Bayes [80]. Their experiments showed that the SVM exhibits the best performance for the attack detection task against the other techniques. Figure 9 reports a detailed picture of the performance achieved. In detail, the results of the proposed SVM were evaluated in terms of accuracy and processing time. Also, the authors tested different

SVM variants using polynomial, normalized polynomial, and RBF kernel types. Polynomial kernel type included five scenarios containing one to five degrees (D1 – D5). According to Figure 9(a), the SVMs equipped with the polynomial kernel provides the highest accuracy, while the RBF-based SVM the lowest. This trend is also confirmed in Figure 9(b) regarding the processing time. Indeed, the polynomial-based kernels provide the lowest (viz. best) processing time compared with that of the other kernel types. Moreover, we can observe that increasing the polynomial degree (i.e. passing from D1 to D5) tends to slightly reduce the accuracy and increase the processing time.




(a) Accuracy




(b) Processing Time

Figure 9. Performance evaluation of SVM proposed in [76] equipped with different kernel types.

Pitropakis et al. [81] provided a network attack detection method that uses the GA presented in [82]. Authors demonstrated that attacks and malicious activities could be identified in a cloud environment by monitoring the system calls produced during the different steps of the attack and comparing the system calls with other executions of the same attack and also with the normal system state when the attack took place.

For the detection of VM-to-hypervisor attacks, Nezarat et al. [83] described a GT-based IDS method in which several agents distinguish the attack and its source using game theory. Besides, the Nash equilibrium concept is leveraged for this purpose. The authors used the attack detection rate to evaluate the performance of the GT-based approach presented and compared it with (i) a model for service-oriented architectures, (ii) parallel neural networks, (iii) genetic algorithms and fuzzy logic, and (iv) genetic algorithms and neural networks. As shown in Figure 10, the proposed GT-based algorithm increased the attack detection rate up to 86%, while both the overhead of the system and the number of false alerts are reduced compared to the other methods.

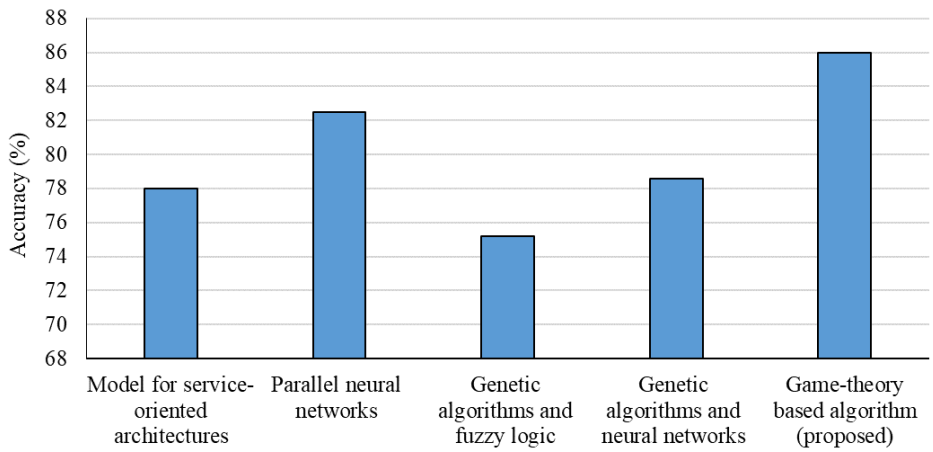



Figure 10. Comparison of the GT-based algorithm proposed in [83] with relevant baselines for the detection of VM-to-hypervisor attacks.

Osanaiye et al. [84] have introduced an ensemble-based method (EMFFS) to reduce the computational complexity and increase the classification precision. To this aim, the devised method is used in the pre-processing step that can eliminate the redundant features to speed up the data classification via a DT. Using this approach, the authors achieved more efficient learning time, lesser complexity, and higher detection rate when discriminating attacks from normal traffic in CC. Results were assessed using the common accuracy measure. Specifically, the accuracy of the proposed method was compared with that obtained with (i) the correlation-based feature selection (CFS) [161], (ii) CFS, consistency-based filter (CONS) and INTERACT [162], (iii) gradual feature removal [163], (iv) consistency subset evaluator (CSE) and CFS [164], and (v) linear correlation-based [165]. According to Figure 11, the DT trained on the feature set extracted by means of the proposed EMFFS method provided the highest accuracy in the detection of DDoS attacks in the CC environment. Moreover, the authors showed that EMFFS can effectively reduce the number of features from 41 to 13, consequently reducing the complexity of the classification task, when compared to the other classification techniques.

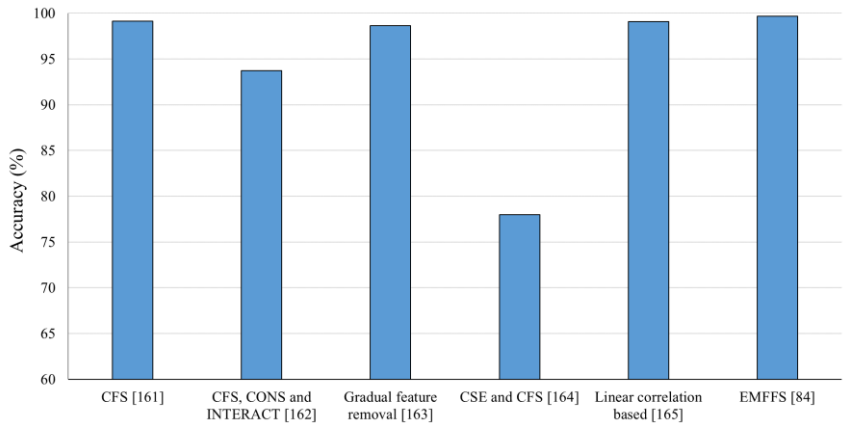



Figure 11. Classification accuracy related to the feature sets extracted via EMFFS [84] and state-of-the-art baselines.

Kumar et al. [85] introduced a cloud IDS based on clustering using learning automata. This technique can enhance the detection performance and it is proposed for healthcare vehicular CC. The learning automata use aggregate relative velocity and connectivity degree to form the leadership of clusters. After this process, the step to secure the data starts. In this step, an automaton utilizes the HMAC algorithm to validate messages. By using the proposed technique, approximately 93% of malicious activities are detected, and adding mobility in leadership formation results in a lower false-positive rate. Moreover, this method can adapt to the changes of the nodes in the network.

Huang et al. [86] proposed an anomaly detection method with the use of LOFs and dimension reasoning rules. By using this technique, the authors can discover the anomalies and their possible origin, being also an effective method for VM management. Specifically, this novel algorithm can detect the behavior of anomalies via VMs' performance profile. By utilizing the proposed method, the rate of detection increases to 98%, while the rate of false alert decreases to 16.9% with respect to the classic LOF used as a baseline.

In [87], the authors introduced a new IDS model by combining PSO and Bayesian networks. This means that quantum behaved-PSO is utilized for learning the Bayesian network structure. The authors used the KDD CUP 99 dataset for their experiment and indicated that the proposed method has effective results in terms of false-positive rate, detection rate, and detection time.

Sharma et al. [88] proposed an IDS approach to detect a DDoS attack with the utilization of an artificial bee colony. This approach consists of three phases: feature selection, artificial bee colony utilization, and decision-making. The goal of the paper is to demonstrate the effectiveness of artificial bee colony technique for (D)DoS attack detection. Results were evaluated using accuracy measure for attack detection. Figure 12 presents the comparison of the devised ABC method with a quantum behaved PSO (QPSO) baseline. According to Figure 12, the ABC method increased the accuracy ($\simeq$+8%) against that obtained with the PSO-based competitor.

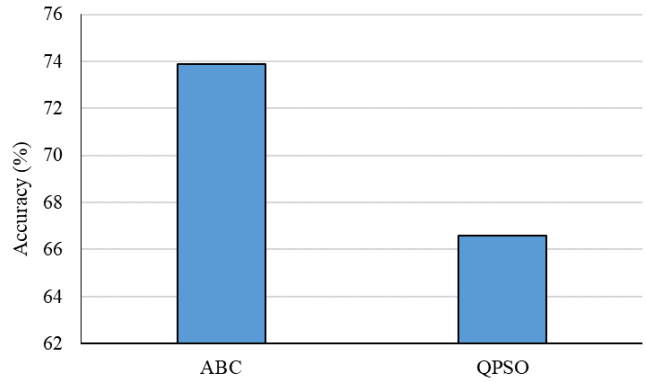



Figure 12. DDos attack detection accuracy of ABC method proposed in [88] against the PSO-based baseline.

Muthukumar and Kumar [89] presented an IDS technique for a private cloud with the help of artificial intelligence. Firstly, they trained the IDS components, secondly performed the test of the trained IDS to see if the training phase completed successfully or not, and thirdly updated the IDS accordingly. The results demonstrated that the new technique could enhance an IDS used in private CC in terms of both time and space complexity.

Chiba et al. [90] introduced a novel NIDS method combining both signature-based and anomaly-based detection, which is an optimized Back-Propagation NN and Snort IDS [153] to detect unknown/known attacks, respectively. At first, Snort examines received packets. If the packet is an intrusion, then Snort sends the alert, otherwise transmits the packet to the anomaly detection phase. In the anomaly detection phase, a back-propagation NN determines the type of packet to check if it is normal or abnormal. Alerts that are generated by the system are saved in a database. Then, the IDS utilizes this database to discover the intrusions. This technique can increase the detection accuracy and minimize the rates of false negatives and false positives and can also guarantee an appropriate cost for computations.

Ghosh et al. [91] proposed a method to minimize the size of the dataset using a combination of Penalty-Reward based instance selection and Nearest Neighbor reduction. To demonstrate the efficacy of this approach, the authors compared NN, AdaBoost, and Random Forest classifiers on the original and reduced dataset. Figure 13 shows the accuracy of the aforementioned methods obtained on both the datasets. It should be noted that the proposed data-optimization technique helps not only to reduce the training time (as an effect of the data-dimensionality reduction) but also to produce better classification accuracy for the designed IDS.

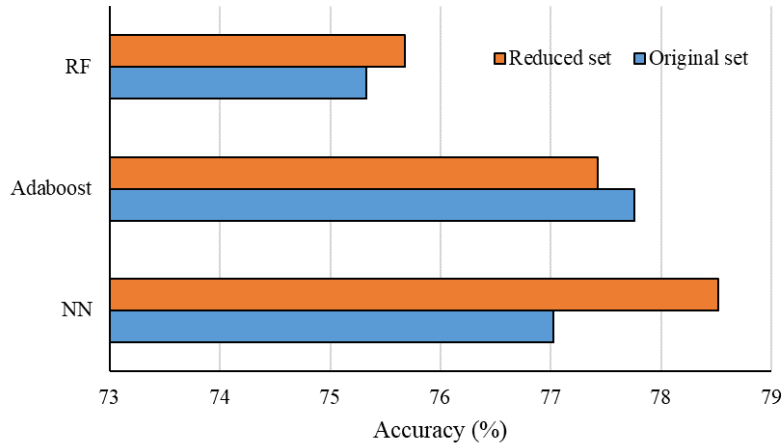



Figure 13. Accuracy of RF, Adaboost, and NN on the original and reduced datasets obtained through the approach presented in [91].

Chonka et al. [92] introduced a defense mechanism based on a back-propagation NN for CC named Cloud Protector. The authors designed this method to discover XML- and HTTP-DoS attacks in a cloud environment. Authors firstly developed their previous service-oriented model SOTA [93, 94] on a cloud system and showed that it can identify the attacks' origin. Then, they trained a NN to detect and filter the DoS attacks. Results demonstrated acceptable values for accuracy and response time, with Cloud Protector being able to discover 98-99% of the XML-DoS traffic within an average of 10-135 ms.

In [95], the authors provided a detection system for cloud and grid computing, using anomaly- and misuse-based detection to verify attacks. In detail, authors employed an ANN for their anomaly-based detection and analyzed communication and log systems' data for misuse-based detection. Their experiment outcomes showed that according to the characteristic of the NN, the false positive is lower than the false-negative rate. Also, the proposed prototype showed low data volume and complexity requirements while providing satisfactory performance for real-time implementation.

Xiong et al. [96] presented an anomaly detection method which analyzes the dynamic characteristics of the network traffic based on catastrophe theory [97] and synergetic NN [98] for a cloud environment. Authors leveraged these two approaches separately, showing that they have both the ability to detect anomalous traffic in the network. Specifically, catastrophe theory can detect unexpected changes in the network traffic and then discover anomalies related to the deviation of the state of the network traffic from the normal one. The synergetic NN is a pattern recognition process that can match the testing data with the training data to perform anomaly detection. Results showed that both methods enhanced the rate of false alerts and detection probability over compared baseline.

Table 8: Cloud IDS methods using single CI techniques. State-of-the-art works are ordered by year.

| ***Work (Year)*** | ***CI Technique*** | ***Aim*** | ***Dataset*** | ***Pros*** | ***Cons*** | ***Cloud Environment*** | ***Contribution*** |
|---|---|---|---|---|---|---|---|
| Vieira et al. [95] (2010) | NN | Intrusion detection | Simulated dataset | • Explores communication events to detect intrusions | • Consumes further time for training | Grid-M [151] | Uses Artificial Neural Network for attack detection |
| Chonka et al. [92] (2011) | NN | Network security | Private dataset | • Detects and filters most of the attacks<br>• Identifies the source of attacks in a short time | • Does not provide the numerical analysis of false alarm rate | Amazon EC2 | Analyzes how X-DoS and H-DoS attacks affect CC |
| Khorshed et al. [76] (2012) | SVM | Proactive attack detection | Simulated dataset | • Detects attack at preparation time<br>• Notifies system admin about the attack type | • Does not consider false alert rate and detection time | Virtual Cloud environment | Detects malware attacks |
| Liu et al. [87] (2013) | PSO | Anomaly detection | DARPA KDD Cup 1999 [109] | • Better convergence speed and detection rate than baselines | • Computational time increases quickly with the number of iterations | *N/A | Performs network anomaly detection using Bayesian quantum PSO |
| Iyengar et al. [74] (2014) | Fuzzy Logic | DdoS detection | Private dataset | • Robust and cooperative DdoS detection method | • Considers only DdoS attacks | Simulated environment | Presents a protection mechanism against DDoS attack and provide a DdoS attack and defense taxonomy |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Pitropakis et al. [81] (2014) | GA | Network-based attack detection | Private dataset | • Good accuracy<br>• Effectiveness with workload increase | • Does not consider detection time and false alarm rate | Cloud with KVM | Detects malicious insider attacks |
| Wang et al. [108] (2014) | Fuzzy Logic | Botnet detection | Private dataset | • Higher detection rate and lower error rate than baselines | • Rate of false alarms is above 1% | Windows Azure | Provides a behavior-based botnet detection |
| Xiong et al. [96] (2014) | NN | Anomaly detection | DARPA KDD Cup 1999 [109] | • Detects anomalies effectively<br>• Low false alarm rate | • Detects only anomalies of the network layer<br>• Does not consider the detection time | *N/A | Analyzes the dynamic characteristics of the network traffic based on the synergetic NN and the catastrophe theory |
| Kumar et al. [85] (2015) | Learning Automata | Intrusion detection | Private dataset | • Adaptive method | • Tested just for two types of attack | Network simulator | Provides a distributed IDS |
| Rajendran et al. [89] (2015) | Muthu-Praveen Algorithm | Intrusion detection | Private dataset | • Detects any type of intrusion within the host as well as in the network | • Tested only in a private cloud environment | Private cloud environment | Identifies the intrusion of unauthorized users in the cloud environment |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Wang et al. [75] (2015) | SVM | Malware detection | Simulated dataset | • Detects zero-day malware<br>• The low false-negative rate<br>• Effective malware detection<br>• Combines misuse detection and anomaly detection | • Cannot handle native code or HTML5-based applications | *N/A | Detects and classifies malware accurately |
| Chan et al. [71] (2016) | Fuzzy Logic | Intrusion detection and prevention system | Simulated dataset | • The high detection accuracy rate<br>• Low false alarm rate | • Does not consider the throughput, latency, and accountability metrics | Public cloud platform with .NET framework 4.5 | Protects SaaS from web service attacks |
| Chiba et al. [90] (2016) | NN | Intrusion detection | Private dataset | • High detection rate<br>• The low false-positive rate<br>• The low false-negative rate<br>• Affordable computational cost<br>• Detects any violation of the security policy | • Does not examine the detection time<br>• Does not provide the numerical results of the experimental evaluation | Virtual cloud environment | Devises a network intrusion detection system with Snort and backpropagation NN that combines signature-based and anomaly-based detection |
| Ghosh et al. [91] (2016) | NN | Intrusion detection | NSL-KDD [110] | • Diminishes the noisy instances as much as possible | • Considers only classification accuracy | *N/A | Employs a data reduction technique for IDSs in CC |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Huang et al. [86] (2016) | LOF | Anomaly detection | Simulated dataset | • Identifies possible sources of the anomaly<br>• High detection rate | • False alarm rate is higher than 1% | Unknown | Detects anomalies over VM live migration |
| Osanaiye et al. [84] (2016) | DT | DDoS detection | NSL-KDD [110] | • Keeps or improves the classification accuracy with a reduced feature set | • Considers only DDoS attacks | *N/A | Provides a feature selection method to pre-process data |
| Sharma et al. [88] (2016) | Artificial Bee Colony | Intrusion detection | Simulated dataset | • Very effective for DDoS detection | • Considers only DoS attacks<br>• Detection rate is under 90% | CloudSim | Proposes intrusion detection for DDoS attack |
| Watson et al. [73] (2016) | SVM | Malware detection | CAIDA | • Better detection accuracy and lower computational cost than competitors | • Ineffective for some malware samples | Cloud testbed based on KVM hypervisor | Provides online anomaly detection method to reach better accuracy |
| Nezarat et al. [83] (2017) | GT | Distributed intrusion detection | DARPA KDD Cup 1999 [109] | • Accelerates the detection process<br>• Reduces the system overhead | • Accuracy is less than 90% | CloudSim | Detects VM-to-hypervisor attacks |

Note: *N/A: Not available in the paper.

### *3.4. Hybrid Methods in CC*

We define hybrid IDS methods those that establish more CI techniques. Table 9 reports in a practical way the works that have applied CI-based hybrid methods in cloud IDSs. To foster a comparison with single methods, we focus on the same aspects (cf. Section 3.3 and Table 8) also in Table 9. To point out the advancement in the state-of-the-art regarding hybrid methods, we have also ordered by year the works in Table 9.

Ganeshkumar and Pandeeswari [99] proposed a hybrid method based on fuzzy systems and NNs for intrusion detection named adaptive neuro-fuzzy inference system (ANFIS). This is an HVI and can detect network-based activities as well as host-based activities without being directly deployed in the VMs. For the performance evaluation of ANFIS, the authors assumed five types of attacks and used the DARPA's KDD Cup dataset. Results were evaluated using precision, recall, and F-measure values. Figure 14 presents the results in terms of ANFIS comparison with Naïve Bayes, NBRF, and ANN for normal connections. We can notice that ANFIS shows a recall comparable with that of other methods but it has higher precision and consequently F-measure (up to ≃ 97). Furthermore, it is designed to be suitable for Big Data applications.

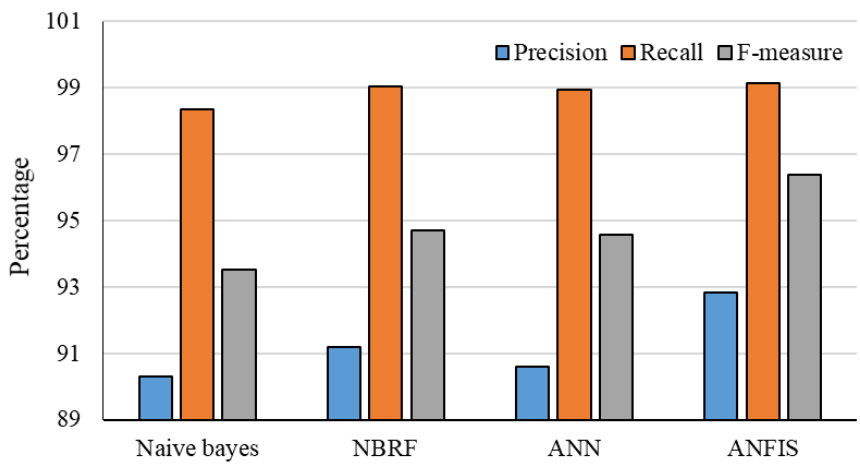



Figure 14. Comparison of ANFIS [99] with baseline methods for attack detection of the DARPA's KDD Cup dataset.

Pandeeswari and Kumar [100] proposed another hybrid method for the hypervisor layer of cloud systems which combines an ANN and Fuzzy C-Means clustering algorithm (FCM-ANN). In this model, there is no need to capture the attack patterns manually. The fuzzy clustering module, ANN module, and fuzzy aggregation module are its three phases. In the first phase, the proposed system groups data into small clusters to improve the ANN's learning ability. In the second phase, the training of ANN modules leverages the values of the defined clusters. The last aggregation module incorporates the outcomes of ANN. Similarly to [99], the authors used DARPA's KDD Cup dataset for the experimental evaluation. Figure 15 reports the precision, recall, and F-measure values resulting from the comparison of the proposed model with Naïve Bayes and standard ANN. FCM-ANN exhibits the highest precision (up to 65%), recall (up to 90%), and F-measure (up to 75%), outperforming Naïve Bayes classifier (showing only a comparable recall) and standard ANN (having the worst performance).

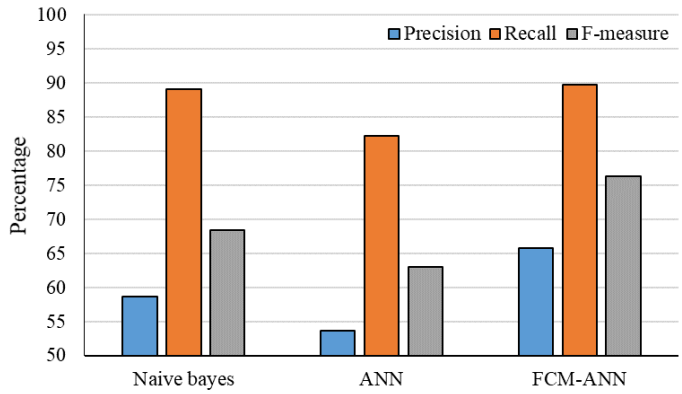



Figure 15. Performance of FCM-ANN [99], Naïve Bayes, and standard ANN on the DARPA's KDD Cup dataset.

Raja and Ramaiah [101] presented an intrusion detection approach with the integration of GA and Fuzzy NN (ANN-GA). The authors used GA to overcome the detection rate problems of Fuzzy NN to distinguish the users-to-root and remote-to-local attacks. Their method includes four levels. In the first level, clustering is applied based on the k-means algorithm [102] which is a proper selection among clustering methods in terms of precision. In the second level, a GA method [103] is utilized to extract the fuzzy rules, and in the third level, this GA is used to optimize the rule base. Finally, in the fourth level, the Fuzzy NN performs the refinement of parameters. After these levels, the authors built a rule base for intrusion detection. Figure 16 presents the results of the hybrid ANN-GA compared with an FNN [166], two variants of the ANN proposed in [167] and [168], and two variant of the GA devised in [169] and [170]. According to the results in Figure 16, obtained using a standard IDS benchmark data, we can notice that the proposed ANN-GA has the best average detection accuracy with respect to the other approaches. Performance in terms of precision, recall, MSE, and scalability shows analogous trends and is not reported for the sake of conciseness.

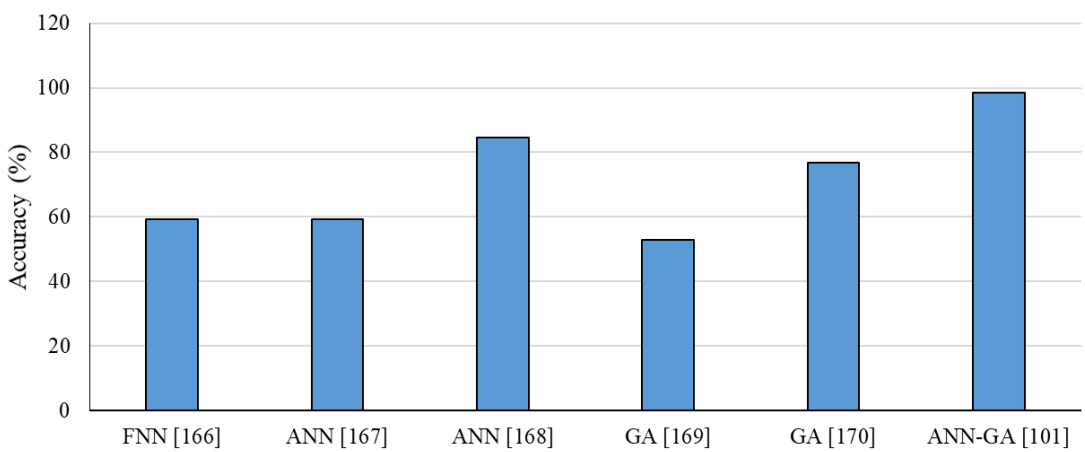



Figure 16. Average detection accuracy of ANN-GN devised in [101] compared with state-of-the-art variants of FNN, ANN, and GA methods.

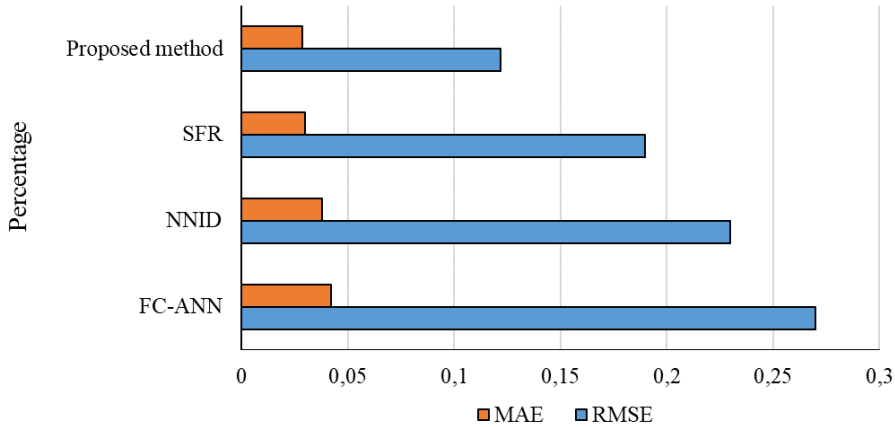



Figure 17. MAE and RMSE of the hybrid method proposed in [138] and state-of-the-art baseline methods.

Ghosh et al. [104] proposed a hybrid IDS combining multi-threaded HIDS and NIDS. The packet analyzer employs a hybrid NN and K-Nearest Neighbor approach (KNN-NN) to analyze the network traffic. Both anomaly and misuse detections are taken into account. At first, the network packets are captured and sent to the analyzing module. For the analysis, the KNN-NN classifier arranges packets into normal or abnormal. Then, an ANN analyzes only the abnormal packets to determine the type of attack. Additionally, the authors used HIDS to detect hypervisor attacks. The combination of HIDS and NIDS leads to a reliable and secure system and this IDS is faster and more efficient than competitors. Indeed, it can handle large flows of data packets, analyze them, and generate reports, improving also the detection accuracy.

In [105], the authors provided an anomaly detection method based on a clustering algorithm to detect abnormal VMs (e.g., corrupted with malicious software and attacking other VMs). To reduce the data dimensionality, the authors devised a feature extraction algorithm based on Locality Preserving Projections [106] and Principal Components Analysis [107]. The actual anomaly detection uses a novel distance-based clustering algorithm fed with the extracted features. The experimental outcomes show that the devised method has higher efficiency in terms of precision, recall, false alarm rate, and runtime.

Idhammad et al. [137] used data mining techniques to design a DIDS for CC. The proposed system includes five modules providing the collection of network traffic, preprocessing of data, detection of anomalies, synchronization of malicious data, and classification of attacks. Each router on the edge of the network collects traffic data and sends them to the preprocessing module that uses a time-based sliding window algorithm to process and normalize the data. Then, the anomaly detection module classifies the network traffic as normal or abnormal employing the Naïve Bayes algorithm. After the first anomaly detection step, for each time window, the malicious traffic on each router side is synchronized to a centralized storage server and finally, a Random Forest classifier is used to detect the type of attack. To evaluate the performance of the proposed IDS, the authors exploited the CIDDS-001 dataset and computed AUC and ROC curves for each edge router. The proposed IDS achieves better accuracy and false-positive rate when compared to the Random Forest classifier, reaching 97% average accuracy, 0.21% average false-positive rate, and also an average running time of 6.23s.

In [138], the authors presented a new IDS method for CC, based on the combination of Artificial Bee Colony, ANN, and fuzzy clustering algorithm. In this IDS method, the fuzzy clustering algorithm is in charge of preparing the homogeneous training subsets to improve the training speed rate. To distinguish between

normal and abnormal traffic data, the IDS embeds an ANN (specifically a Multilayer Perceptron) with the Artificial Bee Colony speeding the determination of the ideal values for weights and biases during the training phase of the network. The metrics used to evaluate this IDS method are root mean square error (RMSE) and mean absolute error (MAE). Figure 17 presents the comparison of the proposed method with (i) an FC-ANN, (ii) a network node intrusion detection (NNID), and (iii) selection of relevant features (SRF). The method proposed in [138] exhibits the lowest MAE and RMSE values, with a 2.23% overall improvement in correctly-classified instances over the considered baselines. Also, the authors affirm that the devised IDS can also increase the kappa statistic in comparison with state-of-the-art methods, with $\simeq$ 0.05 improvement over SRF.

Sharma et al. [139] presented a hybrid IDS based on the WLI-fuzzy clustering and ANN, for the hypervisor level of the cloud environment. At first, the WLI-fuzzy clustering algorithm is utilized to obtain distinctive clusters according to the Euclidean distance. The clustering outcome is given to training the back-propagation ANN employed to identify malicious traffic. Performance analysis leverages false-positive rate, true-positive rate, and accuracy as evaluation parameters. The simulation results, on the DARPA's KDD Cup 1999 dataset, show acceptable performance (up to 97% accuracy) of the developed method against k-means and Fuzzy C-Means baselines.

Table 9: Cloud IDS methods using hybrid (viz. multiple) CI techniques. State-of-the-art works are ordered by year.

| ***Work (Year)*** | ***CI Techniques*** | ***Aim*** | ***Dataset*** | ***Pros*** | ***Cons*** | ***Cloud Environment*** | ***Contribution*** |
|---|---|---|---|---|---|---|---|
| Ghosh et al. [104] (2015) | KNN-NN<br>• K-Nearest Neighbor<br>• ANN | Distributed intrusion detection | NSL-KDD [110] | • Handles large flows of data packets<br>• Integrates anomaly and misuse detection | • Detection accuracy is less than 80%<br>• Does not analyze detection time and false alarm rate | *N/A | Deploys HIDS and NIDS within the cloud-IDS, employing a multi-level classifier made of K-NN for anomaly detection and ANN for misuse detection |
| Lin et al. [105] (2015) | • PCA<br>• Locality Preserving Projections<br>• Clustering | Anomaly detection | * N/A | • Proposes an efficient feature extraction algorithm<br>• Better recall, runtime, and precision than baselines | • False alert rate is higher than 10% | OpenStack platform | Detects the VMs that present abnormal behaviors efficiently employing a feature extraction algorithm for dimensionality reduction of data |
| Ganeshkumar et al. [99] (2016) | ANFIS<br>• Fuzzy<br>• NN | Anomaly detection | DARPA KDD Cup 1999 [109] | • High detection accuracy<br>• Low false-negative rate | • Does not consider detection time | *N/A | Deploys the IDS at the hypervisor level and detects both host-based and network-based attacks |
| Pandeeswari et al. [100] (2016) | FCM-ANN<br>• Fuzzy C-means<br>• ANN | Anomaly detection | DARPA KDD Cup 1999 [109] | • Higher performance for low frequent attacks | • Higher execution time despite better performance | CloudSim | Proposes a detection method able to automatically update the attack database |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Raja and Ramaiah [101] (2016) | NFGA<br>• Neuro<br>• Fuzzy<br>• GA | Intrusion detection | CIDD [111] | • Improved detection rate accuracy, precision, recall, MSE, and scalability | • The reduced speed with fewer cloud nodes | Eucalyptus-built cloud | Evaluates information systems and performs early detection of malicious activities for reducing the security risk |
| Idhammad et al. [137] (2018) | • Naïve Bayes<br>• Random Forest | Distributed intrusion detection | CIDDS-001 [152] | • Better accuracy, false-positive rate, and runtime against Random Forest | • Utilizes a preprocessing algorithm to capture network traffics that may increase the total runtime against other hybrid methods | Google Cloud platform | Devises and deploys on an actual cloud platform a DIDS combining Naïve Bayes and Random Forest classifiers that outperforms the standard Random Forest |
| Sharma et al. [139] (2018) | WLI-ANN<br>• WLI-fuzzy clustering<br>• Back-propagation ANN | Intrusion detection | DARPA KDD Cup 1999 [109] | • Higher true positive rate and accuracy<br>• Lower false alarm rate | • False-positive rate is still higher than 10% | CloudSim | Proposes an HVIDS for CC designed with a hybrid approach that combines WLI-fuzzy clustering and ANN and outperforms standard K-means and Fuzzy C-means in simulation |
| Hajimirzaei and Navimipour [138] (2019) | • Fuzzy clustering<br>• Artificial Bee Colony<br>• ANN | Intrusion detection | NSL-KDD [110] | • Reduces root mean square error and mean absolute error<br>• Improves kappa statistic | • Does not consider runtime<br>• Costly combination of proposed algorithms | CloudSim | Presents a hybrid method based on fuzzy clustering, Artificial Bee Colony, and ANN, outperforming in simulation state-of-the-art methods |

Note: *N/A: Not available in the paper.

### *3.5. Cooperative IDS methods in CC*

In addition to single and hybrid methods, another approach for building IDS in Cloud Computing uses multiple IDSs which cooperate to improve the global accuracy or to reduce the detection time. In this scenario, creating a network of collaborative IDSs that guarantees trustworthy and efficient feedback aggregation is challenging. To this end, Fung and Zhu [158] presented the FACID framework that employs data analytical models and hypothesis testing methods to achieve efficient IDS feedback aggregations. Simulation results confirmed that FACID can reduce the communication overhead as well as the computational resources and memory needed to achieve these results when the number of cooperating IDSs is large, outperforming other heuristic methods.

Authors in [140] proposed a cooperative intrusion detection approach, where they did not assume that every cooperating IDS was trustable. Indeed, untrusted IDSs (not necessarily malicious) can affect the detection of suspicious intrusions in the cloud. The developed framework forms trustworthy and distributed IDSs communities, using game theory and a trustworthiness model based on a threshold related to the accuracy of each IDS. Numerical results demonstrate its effectiveness in terms of false-positive and false-negative rates and cost.

In [141] a distributed IDS is presented, where multiple instances cooperate to counter DoS and DDoS attacks. Specifically, the IDSs exchange alerts and determine if accepting the alerts sent from other IDSs or not. However, the evaluation does not show a significant improvement of accuracy or detection time (conversely it needs little more computational effort compared with Snort [153]), with a focus on improving system reliability to avoid a single point of failure.

Abusitta et al. [142] adopted a proactive approach to allow a real-time cooperative IDS that efficiently exploits the historical IDSs' feedback data. The devised model is based on Deep Learning and uses Stacked Denoising AutoEncoders to reconstruct complete IDSs' feedback from partial feedback. Detection accuracy is improved up to 95% when compared to other state-of-the-art Machine Learning-based methods, such as Multilayer Perceptron, Stacked AutoEncoder, and Variational AutoEncoder on KDD '99.

### *3.6. IDS Methods in MCC*

Most of the existing works in the field of Mobile Cloud Computing targets security problems, especially authentication. Alizadeh et al. [149] presented an extensive overview of security challenges in MCC, focusing on authentication techniques. The authors compared state-of-the-art MCC authentication methods considering five evaluation metrics. The difference between conventional CC and MCC is also highlighted to justify the need for techniques targeting this specific scenario and based on the capabilities and limitations of the MCC environment, other than using the already existing ones. Similarly, the authors in [146] detailed the specific security issues of MCC, discussing the solutions presented in the literature to counter them.

Also, Atre et al. [147] started analyzing the general trends in the mobile market and the challenges of MCC platforms, analyzing various (mobile) cloud service providers and the services they offer. Then, they proposed a monitoring system that helps to decide whether offloading the computation to the cloud on the basis of the effects (beneficial or not) on the mobile battery life. Donald et al. [144] also contributed to analyzing the MCC scenario and highlighted why mobility and energy constraints are the main aspects that contribute to make MCC a different scenario that thus requires the use of different methods and techniques compared to traditional CC, posing limitations, for example, on where intrusion detection should be performed.

Kumar et al. [145] considered security issues in MCC, focusing on the vulnerabilities caused by the different types of access networks, and the risk factors for mobile users. They only mentioned the different technologies and tools employed to enforce security and privacy in MCC.

Abdellaoui et al. [148] focused on user privacy problems suffered when different nodes communicate in the mobile cloud scenario; they additionally proposed a multi-agents system adding intelligence to MCC to overcome privacy and availability issues and to support the computing performance. Authors in [143] also highlighted the peculiarities of MCC scenarios for security issues, especially due to the mobility aspect that is not present in traditional CC, and developed an IDS framework for IaaS-based attack defense.

Authors in [112] presented a malware detection method based on a cloud for mobile devices. GT is used to formulate the malware detection game, which consists of mobile devices that offload their application traces to security servers using access points or base stations in dynamic networks. To improve the detection accuracy, the authors devised a learning scheme that employs the known model of the radio channel to assist the reinforcement learning process in the actual malware detection stage. Simulation results exhibit an increased detection accuracy and a reduced detection delay comparing the proposed scheme with the benchmark strategy.

Damopoulos et al. [113] designed a cloud-based IDS for mobile devices. This framework can perform on both cloud and host devices, irrespectively of the underlying platform. The authors used four anomaly detection mechanisms from the literature and applied a Random Forest classifier as the classification engine of their framework. The evaluation of the framework is based on battery consumption, memory, and CPU usage. It can be seen from the results that the CPU and memory usage for detection mechanisms run in the cloud is lower than that in the host or mobile device. The battery consumption is also reduced for operations in the cloud.

Authors in [134] introduced a framework for the detection and prevention of cyberattacks in MCC using a Deep Learning approach. Results are compared with several other Machine Learning-based approaches on three publicly available datasets, showing that the proposed method achieves higher accuracy, precision, and recall in all the cases.

Gai et al. [135] provided a categorization and a review of intrusion detection techniques for MCC in the context of 5G networks, highlighting the challenges of this scenario. They also introduced a higher-level framework for implementing secure MCC by resorting to IDS techniques in mobile cloud-based 5G networks.

## 4. Discussion

This section presents a discussion regarding the state-of-the-art methods proposed in the literature on CI-based IDS techniques for CC and MCC environments analytically analyzed in the previous sections.

The literature survey we have presented in Section 3, shows that several CI-based methods are employed in the field of CC and MCC security. They are designed to exploit either one single technique or the hybridization of various (single) methods with the aim of taking advantage of the proper combination of multiple CI-based techniques. We have clustered and summarized state-of-the-art works according to the solutions proposed, reporting single and hybrid methods in Tables 8 and 9, respectively. Also, by sorting them by publication year, we highlight how these proposals have evolved during time.

We can notice that while earlier works[1] employ single methods, it is clear that over time the focus has shifted toward hybrid methods, despite over the last three years, few works (e.g., [73, 83]) have still preferred single ones primarily to exploit their lower computational complexity. Indeed, hybrid methods have proven to

[1] We recall that we have surveyed the papers published in the last ten years.

be more suitable for the security tasks considered and provide better results [175, 176] according to different performance metrics taken into account (see later discussion). However, as mentioned before, their major drawback resides in higher execution times [100, 101] that constitute a severe constraint, particularly in the resource-limited MCC environment. Interestingly, more than half of hybrid methods surveyed propose various hybridizations of the ANN, underlining the importance of Neural Networks, being considered, in recent works, the most promising solutions for the design of effective IDSs in CC and MCC scenarios. In addition to hybrid ANNs, Deep Learning architectures (e.g., [134, 142]) are increasingly adopted to deal with the detection of intrusions and attacks in the challenging MCC environment, showing their superior performance with respect to shallow Neural Networks.

Our analysis reveals also that CI-based methods are validated by leveraging both private (viz. self-generated) and public (e.g., [126, 127, 183]) datasets. Unfortunately, these latter were not originally meant to encompass attacks typical of CC and MCC environments, thus limiting the soundness of the results obtained. Also, state-of-the-art solutions—more complex and hybrid methods in particular—are usually tested in simulated environments (e.g., using CloudSim) without proving their effectiveness in an actual (mobile) cloud deployment (as done e.g. in [92, 108, 137]).

To obtain a performance picture of reviewed approaches, in the following, we firstly present an overview of the relevant measures employed to evaluate the effectiveness of the proposed methods, adopting a common nomenclature independent of the specific area of application (e.g., anomaly detection vs. attack/malware classification). Then, we perform a fair comparison based on the most commonly used performance figures.

Figure 18 shows the share of the performance measures utilized in the reviewed studies to evaluate the devised methods. First of all, we can notice that the accuracy is the most frequently-used performance measure (with up to 40% share) for evaluating their effectiveness. Formally, the accuracy is defined as the fraction of correctly classified samples among the total number of samples [172, 173] and is calculated as:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

where TP denotes the true positives, TN the true negatives, FP the false positive, and FN the false negatives.

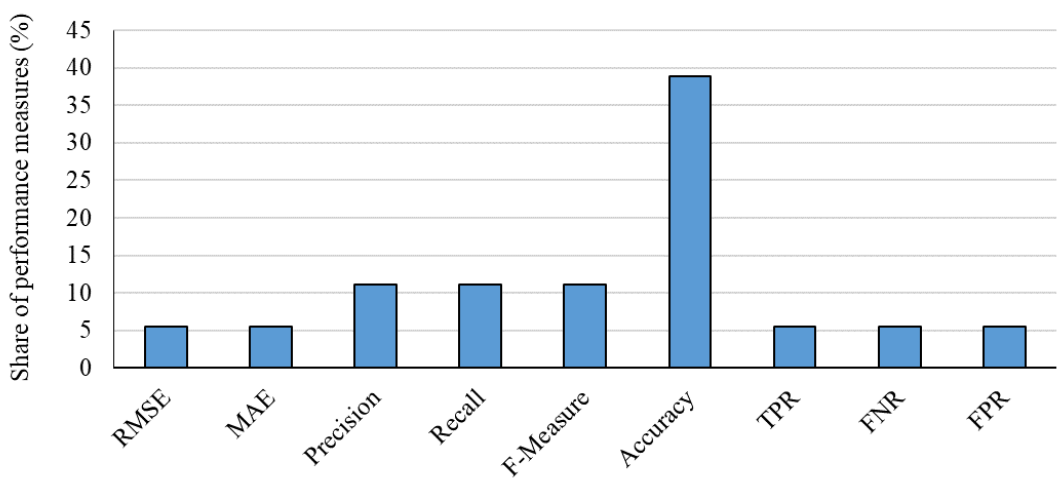



Figure 18. Share of performance measures most-frequently employed in related works.
TPR: True Positive Rate; FNR: False Negative Rate; FPR: False Positive Rate.

However, some works refer to the accuracy as average classification rate in the case of a multi-class attack classification [75] or as attack detection [83] or average detection accuracy [101] in the case of binary attack/intrusion detection.

Per-class (or binary detection) metrics are also used to assess the performance of considered CI-techniques. In addition to the common true-positive, false-negative, and false-positive rates, other measures are borrowed from the Machine Learning domain, namely the precision (prec, i.e. the share of classifier decisions for a certain class that are actually correct) and the recall (rec, i.e. the class-conditional accuracy). The latter two measures are usually combined to account for both their effects concisely, utilizing the F-measure defined as the harmonic mean of precision and recall:

$$F\text{-}measure = (2 \times prec \times rec)/(prec + rec)$$

Additionally, to evaluate the impact of incorrectly-detected intrusions in CC and MCC environments, mean absolute error (MAE) and root mean square error (RMSE) are also employed, being however less frequently used.

According to this outcome, the accuracy has been selected as the relevant comparative measure for the state-of-the-art methods considered. Figure 19 depicts the comparison of the accuracy values of methods developed in the most-relevant state-of-the-art literature. First, we can notice that hybrid methods (e.g., ANN + GA [101], CFS + CONS + INTERACT [162]) provide the highest accuracy compared to the single methods exhibiting a performance drop down to 50% accuracy (e.g., GA [169]).

Dineva et al. [173] have also confirmed this claim as a finding of their review of Machine Learning methods used for the design and control of rotating electrical machines. Similarly, Mosavi et al. [174] compared the performance of different Machine Learning techniques for optimizing energy systems. According to their conclusions, hybrid Machine Learning methods show increased accuracy, robustness, precision, and generalization ability when applied to energy systems. More recently, a multi-classification approach exploting the combination of hybrid Machine Learning classifiers devised for network traffic classification has been proposed in [185]. The authors compared four classes of fusion techniques differing in accepted classifiers' output, training requirements, and learning philosophy. The combination results showed better performance over single techniques according to all considered measures. Nosratabadi et al. [175] surveyed the performance of a hybrid Machine Learning-based method against a single one for handling datasets related to smart cities and sustainable developments. Their outcomes demonstrated the considerable superiority of hybrid techniques over single counterparts in accomplishing the considered task. The motivation for this superiority has been investigated also in [176]. The authors claimed that hybrid methods can finalize the advantages of two or more single methods helping them to overcome their weaknesses and consequently increase their performance.

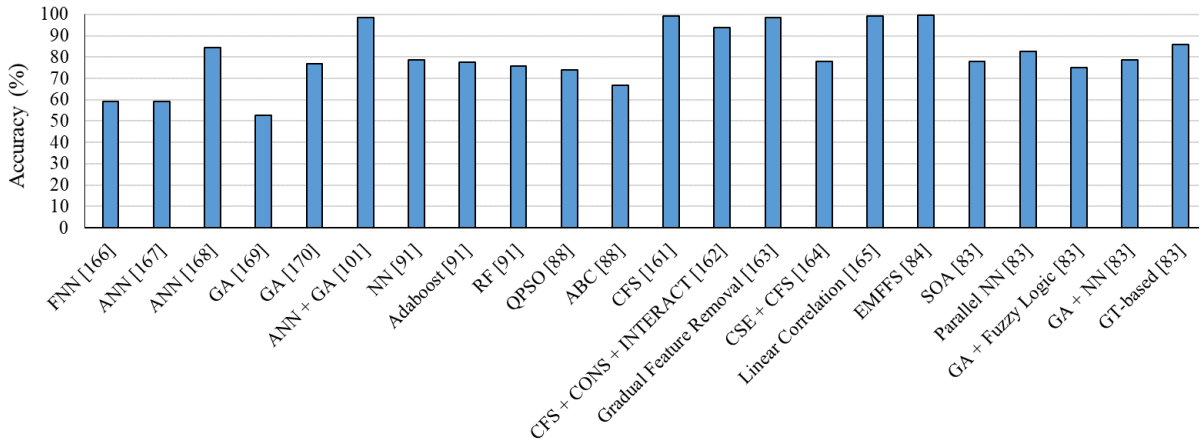



Figure 19. Comparison of state-of-the-art methods in terms of accuracy. + denotes hybrid methods.

To provide a summarizing overview, Table 10 reports selected examples of single and hybrid methods used in both CC and MCC for the design of effective IDSs. In detail, we recall their security issues as well as their main pros and cons in terms of performance achieved by each method taken into account.

Table 10: Selected CI methods used for (M)CC-based IDSs.

| ***Ref.*** | ***Method*** | ***Security Issue*** | ***Performance*** |
|---|---|---|---|
| [73] | SVM | Ineffective for some malware samples | Good detection accuracy and low computational cost |
| [81] | GA | Designed for insider attacks | Good accuracy also with higher workloads |
| [84] | DT | Considers only DoS attacks | Improved classification accuracy with fewer features |
| [88] | Artificial Bee Colony | Considers only DoS attacks | Efficient in the detection of DoS attacks |
| [99] | ANN + Neuro-Fuzzy | Vulnerable to Kernel attacks | Good detection rate and low false-positive rate but missing detection time |
| [100] | ANN + Fuzzy C-means | Vulnerable to Kernel attacks | High detection rate but long runtime |
| [134] | Deep Learning | Needs offline deep training | Outperforms Machine Learning-based and shallow approaches |

## 5. Open Issues for CI-based IDS in CC and MCC

The analyses and discussion presented herein have shown that due to the specific characteristics of the (mobile) cloud environment and its complexities—such as broadband network access, multi-tenancy, resource pooling, rapid elasticity, etc.—providing an effective IDS for such an environment is still challenging. To convey an overview of these challenges, we classify some of the open issues as follows.

**Location of the IDS**: Attacks and intrusions can occur in different layers of cloud, so designing a proper IDS which can detect all types of attacks and malicious activities in different layers of a cloud system is difficult and remains as an open issue in this field. Placing IDSs in each layer and or on every existing VM and hypervisor may cause additional costs and can increase the time of computations. This problem is further exacerbated in MCC because mobile devices are resource-constrained and implementing an IDS on the device may consume more energy and is usually infeasible. Indeed, the tradeoff on deciding what should be offloaded from the mobile device to the cloud, depending on the produced effects, is still an open challenge.

**Lack of updated signatures**: VMs and hypervisors play a vital role in cloud environments and are constantly targeted by new attacks; a detection method assumes these entities too. Unfortunately, the datasets and databases that have been used in discussed detection methods are outdated and may not consider the new attacks' behaviors and signatures. Moreover, since the nature of mobile traffic is different from the fixed network traffic [132], specific mobile-based datasets [154] should be employed for validation and evaluation of IDSs for MCC.

**Wireless technology**: CC and particularly MCC utilize wireless technology to communicate with users' systems. Because of some characteristics of wireless detection such as resource constraints, mobility, and link's limited bandwidth, some issues in terms of management and security remain open.

**Tuning of CI-based IDS methods**: According to papers surveyed in this article, CI-based techniques have been used in anomaly and signature-based intrusion detection systems for (mobile) clouds. If the sensitivity of the attack detection in these IDS methods is not adjusted, then the rate of false alerts may increase. Besides, choosing the right classifiers and features for the detection process is another issue that has been considered by researchers with the integration of novel CI or other algorithms for feature selection and extraction, and classifier tuning. In this way, researchers can provide a fast and more accurate IDS with low false-negative rate and high true positive rate in CC environments. However, the classifiers and related features designed for CC could not be suitable for intrusion detection in MCC, thus more effort is needed to continue to fill this gap.

**Performance-evaluation criteria selection**: The kind of deployed environment and the databases used are two instances of possible choices for performance-evaluation criteria. The definition of the evaluation setup can be a severe open issue for cloud and mobile cloud security. Sometimes the approaches which have been presented to enhance the performance of an IDS may not be efficient under some conditions or it may lead to an increase in the rate of the false alerts because of an unsuitable selection of performance criteria.

**Multi-tenancy**: One of the characteristics of CC/MCC is sharing resources and services between cloud users in a multi-tenant environment. Thus, sharing of services or data between several users being the tenants of a cloud service leads to storing users' information on the same machines on which virtual resources are deployed. The integrity and privacy of information maintained by cloud providers and possible vulnerabilities leading to data leaks remain a challenge in this field.

**Security policy**: Since VMs are added and deleted dynamically, the security policy changes continuously. A cloud service provider is responsible for creating an appropriate and customizable security policy for CC, which is a challenging task. Because of more distributed data, multiple storage resources, and distributed infrastructure of MCC, this task is more complicated in this latter scenario and also leads to spending more time and resources for effective intrusion detection [121].

**Heterogeneity of environment**: The heterogeneity aspect is much more urgent in the MCC environment. Indeed, in MCC mobile devices utilize various types of wireless network interfaces for connection to the network. This heterogeneity causes different problems for IDSs and makes its implementation more complex. The IDS must adapt itself to different response mechanisms or functionalities of devices in the network and its database must be constantly updated since the fast-paced evolution of mobile traffic.

According to open issues and challenges discussed, an IDS can be considered suitable when it manages to cope with all aspects and characteristics of cloud and mobile cloud and it can provide comprehensive protection against different types of known/unknown attacks in different layers of cloud or mobile devices.

## 6. Conclusion and Future Perspectives

In this paper, we have presented a survey of intrusion detection techniques based on computational intelligence applied in both cloud and mobile cloud computing environments.

At first, we have introduced a brief review of CC, MCC, and IDS to provide the context of our analysis. CC refers to computations that are run by several remote servers which are connected by a network that leads to centralized data storage and online access to computer resources and services. MCC is a new paradigm derived from CC and mobile computing in which mobile devices with limited resources can offload their complex computations to a cloud server. Cloud services are accessible via the Internet so security and privacy of transferred data between users and cloud is crucial and should be provided. Due to the increasing use of CC/MCC services, cybersecurity attacks are also raised in this environment. To protect the latter against various inside and outside attacks, an IDS can be a good option. An IDS is a hardware or software device that can automatically alert network admin when a malicious activity or security violation occurs.

Then, we have found and studied papers that use CI techniques for intrusion detection, and we have provided a comprehensive survey of these methods. Surveying the literature, we have identified different classifications of IDS techniques with the most known types being misuse-based and anomaly-based. On the basis of reviewed articles, we have found that the VMI and HVI are two useful types of IDSs for the cloud environment and can provide the best performance in intrusion detection. CI techniques utilized in IDSs can further increase the accuracy of detection and can decrease the false alert rates. Thus, the focus of this paper has been on the usage of CI-based IDS in cloud environments. Specifically, we have classified CI techniques into single methods and hybrid methods to determine the advantages and disadvantages of each group. Cooperative CI-based IDS methods and MCC-specific solutions have had dedicated discussions. Also, we have spent some words on the most common attacks and the datasets used for validation and performance evaluation of intrusion detection techniques. Open issues in cloud and mobile cloud IDSs are further pointed out and discussed.

With our work, we have been able to highlight how CI-based IDSs require careful tuning of the parameters to reach their goal with high accuracy; moreover, the performance-evaluation criteria adopted across different works are not always uniform. In addition, intrusion detection datasets are still lacking, and most of the works base their evaluation on KDD ’99 and its variants, thus referring to a 20 years old dataset not suitable for the validation of IDS proposals in the (mobile) cloud environment. The heterogeneity and dynamicity of CC (and even more of MCC) poses an additional challenge, as signature-based approaches require a constant update of the knowledge base which is hard to obtain. Also, the location where to deploy an IDS may differ according to the application and is also still subject for further research.

Finally, while CC and MCC are similar paradigms, there is a lack of proper IDS in MCC scenarios; indeed, most of the researchers focused on authentication methods to secure data in MCC. This shortage of works can be due to the applicability of existing CC-IDSs for MCC, the inefficiency of these methods for MCC, or because of the lack of datasets and common performance criteria to use in mobile scenarios. Given these

limitations, we argue that more research for CI-based IDS in MCC is needed. Also, from our discussion, further *future directions* can be envisioned. First, there is a need for a common performance evaluation benchmark that takes into account (i) well-defined metrics, (ii) a unified CC/MCC platform, and (iii) common affecting factors, to make it possible to obtain a comparable assessment of different proposals. Besides, security issues in similar scenarios, such as software-defined CC [187], mobile edge computing, and fog computing, should be explored. However, moving toward different domains would require datasets that are able to catch the peculiarities of the considered environments in order to provide meaningful results. Cutting-edge solutions successfully applied in other areas can also contribute to increasing the efficacy of CI-based IDSs in CC/MCC, namely: Explainable AI [188], Deep Reinforcement Learning and Big Data [189], and advanced collaborative IDS paradigms (e.g., Deep Learning-based collaborative IDS).

## Acknowledgments

This work has been supported by the Norwegian Open AI Lab.

## References

[1] P. Mell and T. Grance, The NIST definition of cloud computing, 2011.
[2] K. Chandrasekaran, Essentials of Cloud Computing: CRC Press, 2014.
[3] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, A survey of intrusion detection techniques in cloud, Journal of Network and Computer Applications, Vol. 36, No. 1, pp. 42-57, January, 2013.
[4] A. Patel, M. Taghavi, K. Bakhtiyari, and J. C. JúNior, An intrusion detection and prevention system in cloud computing: A systematic review, Journal of Network and Computer Applications, Vol. 36, No. 1, pp. 25-41, January, 2013.
[5] N. Siddique and H. Adeli, Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing: John Wiley & Sons, 2013.
[6] A. P. Engelbrecht, Computational Intelligence: An Introduction: John Wiley & Sons, 2007.
[7] C. N. Modi and K. Acha, Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review, The Journal of Supercomputing, Vol. 73, No. 3, pp. 1192-1234, March, 2017.
[8] S. Iqbal, M. L. M. Kiah, B. Dhaghighi, M. Hussain, S. Khan, M. K. Khan, et al., On cloud security attacks: A taxonomy and intrusion detection and prevention as a service, Journal of Network and Computer Applications, Vol. 74, pp. 98-120, October, 2016.
[9] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, Distributed denial of service (DdoS) resilience in cloud: review and conceptual cloud DdoS mitigation framework, Journal of Network and Computer Applications, Vol. 67, pp. 147-165, May, 2016.
[10] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, Intrusion detection techniques in cloud environment: A survey, Journal of Network and Computer Applications, Vol. 77, pp. 18-47, January, 2017.
[11] U. Kumar and B. N. Gohil, A survey on intrusion detection systems for cloud computing environment, International Journal of Computer Applications, Vol. 109, No. 1, pp. 6-15, January, 2015.
[12] R. Denz and S. Taylor, A survey on securing the virtual cloud, Journal of Cloud Computing: Advances, Systems and Applications, Vol. 2, No. 1, pp. 17, November, 2013.
[13] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, Mobile cloud computing: A survey, state of art and future directions, Mobile Networks and Applications, Vol. 19, No. 2, pp. 133-143, April, 2014.
[14] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, Heterogeneity in mobile cloud computing: taxonomy and open challenges, IEEE Communications Surveys & Tutorials, Vol. 16, No. 1, pp. 369-392, First Quarter, 2014.
[15] H. Suo, Z. Liu, J. Wan, and K. Zhou, Security and privacy in mobile cloud computing, in 9th Int. Wireless Communications and Mobile Computing Conf., Sardinia, Italy, 2013, pp. 655-659.
[16] A. N. Khan, M. M. Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: A survey," Future Generation Computer Systems, Vol. 29, No. 5, pp. 1278-1299, July, 2013.
[17] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, Security and privacy challenges in mobile cloud computing: Survey and way ahead, Journal of Network and Computer Applications, Vol. 84, pp. 38-54, April, 2017.
[18] N. Fernando, S. W. Loke, and W. Rahayu, Mobile cloud computing: A survey, Future Generation Computer Systems, Vol. 29, No. 1, pp. 84-106, January, 2013.
[19] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, A survey of mobile cloud computing: architecture, applications, and approaches, Wireless Communications and Mobile Computing, Vol. 13, No. 18, pp. 1587-1611, December, 2013.
[20] L. Guan, X. Ke, M. Song, and J. Song, A survey of research on mobile cloud computing, in 10th IEEE/ACIS Int. Conf. on Computer and Information Science, Sanya, China, 2011, pp. 387-392.
[21] Y. Xu and S. Mao, A survey of mobile cloud computing for rich media applications, IEEE Wireless Communications, Vol. 20, No. 3, pp. 46-53, June, 2013.

[22] Y. Wang, R. Chen, and D.-C. Wang, A survey of mobile cloud computing applications: perspectives and challenges, Wireless Personal Communications, Vol. 80, No. 4, pp. 1607-1623, February, 2015.
[23] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, et al., NIST cloud computing reference architecture, NIST Special Publication 500-292, September, 2011.
[24] Microsoft Azure. Available: https://azure.microsoft.com/en-us/
[25] Amazon EC2. Elastic Compute Cloud. Available: https://aws.amazon.com/ec2/
[26] Google AppEngine. Available: https://cloud.google.com/appengine/
[27] IBM's Blue Cloud. Available: https://www.ibm.com/cloud/
[28] Y. Huang, H. Su, W. Sun, J. M. Zhang, C. Guo, J. Xu, et al., Framework for building a low-cost, scalable, and secured platform for Web-delivered business services, IBM Journal of Research and Development, Vol. 54, No. 6, pp. 4:1-4:14, November-December, 2010.
[29] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems, Vol. 25, No. 6, pp. 599-616, June, 2009.
[30] G. Huerta-Canepa and D. Lee, A virtual cloud computing provider for mobile devices, in Proc. Of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, San Francisco, CA, 2010, pp. 1-5.
[31] D. Kwon, S. Yang, Y. Paek, and K. Ko, Optimization techniques to enable execution offloading for 3D video games, Multimedia Tools and Applications, Vol. 76, No. 9, pp. 11347-11360, May, 2017.
[32] S.-P. Chuah, N.-M. Cheung, and C. Yuen, Layered coding for mobile cloud gaming using scalable blinn-phong lighting, IEEE Transactions on Image Processing, Vol. 25, No. 7, pp. 3112-3125, July, 2016.
[33] J.-H. Yang, An Electronic Transaction Mechanism Using Mobile Devices for Cloud Computing, Wireless Personal Communications, Vol. 94, No. 3, pp. 713-724, June, 2017.
[34] X. Yang, T. Pan, and J. Shen, On 3G mobile e-commerce platform based on cloud computing, in 3rd IEEE International Conference on Ubi-media Computing, Jinhua, China, 2010, pp. 198-201.
[35] D. B. Hoang and L. Chen, Mobile cloud for assistive healthcare (MoCAsH), in IEEE Asia-Pacific Services Computing Conference, Hangzhou, China, 2010, pp. 325-332.
[36] A. T. Lo'ai, R. Mehmood, E. Benkhlifa, and H. Song, Mobile cloud computing model and big data analysis for healthcare applications, IEEE Access, Vol. 4, pp. 6171-6180, 2016.
[37] J. Hanen, Z. Kechaou, and M. B. Ayed, An enhanced healthcare system in mobile cloud computing environment, Vietnam Journal of Computer Science, Vol. 3, No. 4, pp. 267-277, November, 2016.
[38] M. Al-Emran, H. M. Elsherif, and K. Shaalan, Investigating attitudes towards the use of mobile learning in higher education, Computers in Human Behavior, Vol. 56, pp. 93-102, March, 2016.
[39] H. Crompton, D. Burke, K. H. Gregory, and C. Gräbe, The use of mobile learning in science: a systematic review, Journal of Science Education and Technology, Vol. 25, No. 2, pp. 149-160, April, 2016.
[40] J. M. Zydney and Z. Warner, Mobile apps for science learning: Review of research, Computers & Education, Vol. 94, pp. 1-17, March, 2016.
[41] V. Cardellini, V. D. N. Personé, V. Di Valerio, F. Facchinei, V. Grassi, F. L. Presti, et al., A game-theoretic approach to computation offloading in mobile cloud computing, Mathematical Programming, Vol. 157, No. 2, pp. 421-449, June, 2016.
[42] S. Guo, B. Xiao, Y. Yang, and Y. Yang, Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing, in IEEE INFOCOM, San Francisco, CA, 2016, pp. 1-9.
[43] N. I. M. Enzai and M. Tang, A heuristic algorithm for multi-site computation offloading in mobile cloud computing, Procedia Computer Science, Vol. 80, pp. 1232-1241, 2016.
[44] J. Wack, K. Cutler, and J. Pole, Guidelines on firewalls and firewall policy, BOOZ-ALLEN AND HAMILTON INC MCLEAN VA, 2002.
[45] S. Karen and H. Paul, Guidelines on firewalls and firewall policy, NIST Recommendations, Special Publication 800-41, 2008.
[46] A. Patcha and J.-M. Park, An overview of anomaly detection techniques: Existing solutions and latest technological trends, Computer Networks, Vol. 51, No. 12, pp. 3448-3470, August, 2007.
[47] R. Bace and P. Mell, NIST special publication on intrusion detection systems, BOOZ-ALLEN AND HAMILTON INC MCLEAN VA, 2001.
[48] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, Intrusion detection system: A comprehensive review, Journal of Network and Computer Applications, vol. 36, No. 1, pp. 16-24, January, 2013.
[49] P. Stavroulakis and M. Stamp, Handbook of Information and Communication Security: Springer Science & Business Media, 2010.
[50] F. Sabahi and A. Movaghar, Intrusion detection: A survey, in 3rd International Conference on Systems and Networks Communications, Sliema, Malta, 2008, pp. 23-26.
[51] M. Derfouf, M. Eleuldj, S. Enniari, and O. Diouri, Smart Intrusion Detection Model for the Cloud Computing, in Europe and MENA Cooperation Advances in Information and Communication Technologies, ed: Springer, Cham, 2017, pp. 411-421.
[52] P. Deshpande, S. Sharma, S. Peddoju, and S. Junaid, HIDS: A host based intrusion detection system for cloud computing environment, International Journal of System Assurance Engineering and Management, Vol. 9, No. 3, pp. 567-576, June, 2018.
[53] Z. Wang and Y. Zhu, A centralized HIDS framework for private cloud, in 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Kanazawa, Japan, 2017, pp. 115-120.
[54] V. Mahajan and S. K. Peddoju, Deployment of Intrusion Detection System in Cloud: A Performance-Based Study, in IEEE Trustcom/BigDataSE/ICESS, Sydney, Australia, 2017, pp. 1103-1108.

[55] Z. Salek and F. M. Madani, Multi-level Intrusion detection system in cloud environment based on trust level, in 6th International Conference on Computer and Knowledge Engineering, Mashhad, Iran, 2016, pp. 94-99.
[56] V. Balamurugan and R. Saravanan, Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation, Cluster Computing, pp. 1-13, September, 2017.
[57] S. Ram, Secure cloud computing based on mutual intrusion detection system, International Journal of Computer Application, Vol. 1, No. 2, pp. 57-67, February, 2012.
[58] S. Velliangiri and J. Premalatha, Intrusion detection of distributed denial of service attack in cloud, Cluster Computing, pp. 1-9, September, 2017.
[59] S. VivinSandar and S. Shenai, Economic denial of sustainability (edos) in cloud services using http and xml based ddos attacks, International Journal of Computer Applications, Vol. 41, No. 20, March, 2012.
[60] H. R. Ghorbani and M. R. Hashemi, An Improved Distributed Intrusion Detection Architecture for Cloud Computing, in International Symposium on Computer Networks and Distributed Systems, 2013, pp. 105-116.
[61] J. Nikolai and Y. Wang, Hypervisor-based cloud intrusion detection system, in International Conference on Computing, Networking and Communications, Honolulu, USA, 2014, pp. 989-993.
[62] F. Lombardi and R. Di Pietro, Secure virtualization for cloud computing, Journal of Network and Computer Applications, Vol. 34, No. 4, pp. 1113-1122, July, 2011.
[63] H. R. Ghorbani and R. S. Shahrezaie, Toward a policy-based distributed intrusion detection system in cloud computing using data mining approaches, in International Congress on Technology, Communication and Knowledge, Mashhad, Iran, 2015, pp. 412-419.
[64] W. A. Jansen, Cloud hooks: Security and privacy issues in cloud computing, in 44th Hawaii International Conference on System Sciences, Kauai, USA, 2011, pp. 1-10.
[65] L. Jia, M. Zhu, and B. Tu, T-VMI: Trusted Virtual Machine Introspection in Cloud Environments, in Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Madrid, Spain, 2017, pp. 478-487.
[66] S. Bharadwaja, W. Sun, M. Niamat, and F. Shen, Collabra: a xen hypervisor based collaborative intrusion detection system, in 8th International Conference on Information Technology: New Generations, Las Vegas, USA, 2011, pp. 695-700.
[67] J. Shi, Y. Yang, C. Li, and X. Wang, Spems: A stealthy and practical execution monitoring system based on vmi, in International Conference on Cloud Computing and Security, ed: Springer, 2015, pp. 380-389.
[68] T. Garfinkel and M. Rosenblum, A Virtual Machine Introspection Based Architecture for Intrusion Detection, In Ndss, Vol. 3, No. 2003, pp. 191-206, 2003.
[69] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, VAED: VMI-assisted evasion detection approach for infrastructure as a service cloud, Concurrency and Computation: Practice and Experience, Vol. 29, No. 12, March, 2017.
[70] J. Shi, Y. Yang, and C. Tang, Hardware assisted hypervisor introspection, SpringerPlus, Vol. 5, No. 1, p. 647, May, 2016.
[71] G.-Y. Chan, F.-F. Chua, and C.-S. Lee, "Intrusion detection and prevention of web service attacks for software as a service: Fuzzy association rules vs fuzzy associative patterns," Journal of Intelligent & Fuzzy Systems, Vol. 31, No. 2, pp. 749-764, July, 2016.
[72] K. Wang, C.-Y. Huang, L.-Y. Tsai, and Y.-D. Lin, Behavior-based botnet detection in parallel, Security and Communication Networks, Vol. 7, No. 11, pp. 1849-1859, November, 2014.
[73] M. R. Watson, A. K. Marnerides, A. Mauthe, and D. Hutchison, Malware detection in cloud computing infrastructures, IEEE Transactions on Dependable and Secure Computing, Vol. 13, No. 2, pp. 192-205, March, 2016.
[74] N. C. S. Iyengar, A. Banerjee, and G. Ganapathy, A fuzzy logic based defense mechanism against distributed denial of service attack in cloud computing environment, International Journal of Communication Networks and Information Security, Vol. 6, No. 3, pp. 233-245, December, 2014.
[75] X. Wang, Y. Yang, and Y. Zeng, Accurate mobile malware detection and classification in the cloud, SpringerPlus, Vol. 4, No. 1, p. 583, 2015.
[76] M. T. Khorshed, A. S. Ali, and S. A. Wasimi, A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing, Future Generation Computer Systems, Vol. 28, No. 6, pp. 833-851, June, 2012.
[77] E. Frank and I. H. Witten, Generating accurate rule sets without global optimization, 1998.
[78] A. Dainotti, F. Gargiulo, L. I. Kuncheva, A. Pescapè, and C. Sansone, "Identification of traffic flows hiding behind TCP port 80," 2010 IEEE International Conference on Communications. IEEE, 2010.
[79] R. Lopez and E. Oñate, A variational formulation for the multilayer perceptron, in International Conference on Artificial Neural Networks, ed: Springer, pp. 159-168, 2006.
[80] G. H. John and P. Langley, Estimating continuous distributions in Bayesian classifiers, in Proceedings of the 11th conference on Uncertainty in artificial intelligence, Montreal, Canada,1995, pp. 338-345.
[81] N. Pitropakis, D. Anastasopoulou, A. Pikrakis, and C. Lambrinoudakis, If you want to know about a hunter, study his prey: detection of network based attacks on KVM based cloud environments, Journal of Cloud Computing, Vol. 3, No. 1, p. 20, December, 2014.
[82] T. F. Smith and M. S. Waterman, Identification of common molecular subsequences, Journal of Molecular Biology, Vol. 147, pp. 195-197, 1981.
[83] A. Nezarat and Y. Shams, A game theoretic-based distributed detection method for VM-to-hypervisor attacks in cloud environment, The Journal of Supercomputing, Vol. 73, No. 10, pp. 4407-4427, October, 2017.
[84] O. Osanaiye, H. Cai, K.-K. R. Choo, A. Dehghantanha, Z. Xu, and M. Dlodlo, Ensemble-based multi-filter feature selection method for DdoS detection in cloud computing, EURASIP Journal on Wireless Communications and Networking, Vol. 2016, No. 1, p. 130, December, 2016.

[85] N. Kumar, J. P. Singh, R. S. Bali, S. Misra, and S. Ullah, An intelligent clustering scheme for distributed intrusion detection in vehicular cloud computing, Cluster Computing, Vol. 18, No. 3, pp. 1263-1283, September, 2015.
[86] T. Huang, Y. Zhu, Y. Wu, S. Bressan, and G. Dobbie, Anomaly detection and identification scheme for VM live migration in cloud infrastructure, Future Generation Computer Systems, Vol. 56, pp. 736-745, March, 2016.
[87] Y. Liu and R. Ma, Network anomaly detection based on BQPSO-BN algorithm, IETE Journal of Research, Vol. 59, pp. 334-342, 2013.
[88] S. Sharma, A. Gupta, and S. Agrawal, An Intrusion Detection System for Detecting Denial-of-Service Attack in Cloud Using Artificial Bee Colony, in Proceedings of the International Congress on Information and Communication Technology, India, 2016, pp. 137-145.
[89] M. B. and P. K. Rajendran, Intelligent Intrusion Detection System for Private Cloud Environment, in Proceedings of the 3rd International Symposium Security in Computing and Communications, ed Springer, 2015, pp. 54-65.
[90] Z. Chiba, N. Abghour, K. Moussaid, A. E. omri, and M. Rida, A Cooperative and Hybrid Network Intrusion Detection Framework in Cloud Computing Based on Snort and Optimized Back Propagation Neural Network, Procedia Computer Science, Vol. 83, 2016, pp. 1200-1206.
[91] P. Ghosh, A. Saha, and S. Phadikar, Penalty-Reward Based Instance Selection Method in Cloud Environment Using the Concept of Nearest Neighbor, Procedia Computer Science, Vol. 89, 2016, pp. 82-89.
[92] A. Chonka, Y. Xiang, W. Zhou, and A. Bonti, Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks, Journal of Network and Computer Applications, Vol. 34, No. 4, pp. 1097-1107, July, 2011.
[93] A. Chonka, W. Zhou, and Y. Xiang, Defending grid web services from xdos attacks by sota, in IEEE International Conference on Pervasive Computing and Communications, TX, USA, 2009, pp. 1-6.
[94] A. Chonka, W. Zhou, and Y. Xiang, Protecting web services from DdoS attacks by SOTA, in Proceedings of the 5th International Conference on Information Technology and Applications, Macquarie Scientific Publishing, Bathurst, N.S.W., 2008, pp. 379-384.
[95] K. Vieira, A. Schulter, C. Westphall, and C. Westphall, Intrusion Detection for Grid and Cloud Computing, IT Professional, Vol. 12, pp. 38-43, 2010.
[96] W. Xiong, H. Hu, N. Xiong, L. T. Yang, W.-C. Peng, X. Wang, et al., Anomaly secure detection methods by analyzing dynamic characteristics of the network traffic in cloud communications, Information Sciences, Vol. 258, pp. 403-415, February, 2014.
[97] T. Poston and I. Stewart, Catastrophe theory and its applications: Courier Corporation, 2014.
[98] H. Haken, Synergetic computers and cognition: A top-down approach to neural nets, Springer Science & Business Media, 2013.
[99] P. Ganeshkumar and N. Pandeeswari, Adaptive neuro-fuzzy-based anomaly detection system in cloud, International Journal of Fuzzy Systems, Vol. 18, No. 3, pp. 367-378, June, 2016.
[100] N. Pandeeswari and G. Kumar, Anomaly detection system in cloud environment using fuzzy clustering based ANN, Mobile Networks and Applications, Vol. 21, No. 3, pp. 494-505, June, 2016.
[101] S. Raja and S. Ramaiah, An Efficient Fuzzy-Based Hybrid System to Cloud Intrusion Detection, International Journal of Fuzzy Systems, Vol. 19, No. 1, pp. 1-16, February, 2016.
[102] M.-C. Su and C.-H. Chou, A modified version of the K-means algorithm with a distance based on cluster symmetry, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 6, pp. 674–680, June, 2001.
[103] P. Vivekanandan, M. Rajalakshmi, and R. Nedunchezhian, An intelligent genetic algorithm for mining classification rules in large datasets, Computing and Informatics, Vol. 32, No. 1, pp. 1-22, 2013.
[104] P. Ghosh, A. K. Mandal, and R. Kumar, An Efficient Cloud Network Intrusion Detection System, in Information Systems Design and Intelligent Applications, ed: Springer, 2015, pp. 91-99.
[105] M. Lin and S. Chen, An Efficient Anomaly Detection Framework for Cloud Computing Environment, JCP, Vol. 10, No. 3, pp. 155-165, May, 2015.
[106] X. He and P. Niyogi, Locality preserving projections, in Advances in neural information processing systems, 2004, pp. 153-160.
[107] H. Abdi and L. J. Williams, Principal component analysis, Wiley Interdisciplinary Reviews: Computational Statistics, Vol. 2, No. 4, pp. 433-459, July-August, 2010.
[108] K. Wang, C. Y. Huang, L. Y. Tsai, and Y. D. Lin, Behavior-based botnet detection in parallel, Security and Communication Networks, Vol. 7, No. 11, pp. 1849-1859, November, 2014.
[109] N. Chandolikar and V. Nandavadekar, Selection of relevant feature for intrusion attack classification by analyzing KDD Cup 99, MIT International Journal of Computer Science & Information Technology, Vol. 2, No. 2, pp. 85-90, August, 2012.
[110] S. Revathi and A. Malathi, A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection, International Journal of Eng. Research and Technology, Vol. 2, No, 12, pp. 1848-1853, December, 2013.
[111] H. A. Kholidy and F. Baiardi, Cidd: A cloud intrusion detection dataset for cloud computing and masquerade attacks, in 9th International Conference on Information Technology: New Generations, Las Vegas, USA, 2012, pp. 397-402.
[112] L. Xiao, Y. Li, X. Huang, and X. Du, Cloud-based Malware Detection Game for Mobile Devices with Offloading, IEEE Transactions on Mobile Computing, Vol. 16, No. 10, pp. 2742-2750, October, 2017.
[113] D. Damopoulos, G. Kambourakis, and G. Portokalidis, The best of both worlds: a framework for the synergistic operation of host and cloud anomaly-based IDS for smartphones, in Proceedings of the 7th European Workshop on System Security, Amsterdam, The Netherlands, 2014, p. 6.
[114] M. Darwish, A. Ouda, and L. F. Capretz, Cloud-based DdoS attacks and defenses, in International Conference on Information Society, Toronto, Canada, 2013, pp. 67-71.

[115] C. Ambedkar and V. K. Babu, Detection of probe attacks using machine learning techniques, International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), Vol. 2, No. 3, pp. 25-29, March, 2015.
[116] V. R. Vemuri, Detecting And Visualizing Denial-of-Service And Network Probe Attacks Using Principal Component Analysis First Author: Khaled Labib Affiliation: Ph. D. Student, Dept. of Applied Science, University of California, Davis, USA, 2004.
[117] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, Cross-VM side channels and their use to extract private keys, in Proceedings of the 2012 ACM conference on Computer and communications security, North Carolina, USA, 2012, pp. 305-316.
[118] N. Rakotondravony, B. Taubmann, W. Mandarawi, E. Weishäupl, P. Xu, B. Kolosnjaji, M. Protsenko, H. de Meer, H.P. Reiser, Classifying malware attacks in IaaS cloud environments, Journal of Cloud Computing, Vol. 6, p. 26, December, 2017.
[119] A. Milenkoski, B. D. Payne, N. Antunes, M. Vieira, and S. Kounev, Hinjector: injecting hypercall attacks for evaluating VMI-based intrusion detection systems, in Poster Reception at the 2013 Annual Computer Security Applications Conference, 2013.
[120] M. La Polla, F. Martinelli, and D. Sgandurra, A survey on security for mobile devices, IEEE communications surveys & tutorials, Vol. 15, No. 1, pp. 446-471, First Quarter, 2013.
[121] Z. Inayat, A. Gani, N. B. Anuar, S. Anwar, and M. K. Khan, Cloud-Based Intrusion Detection and Response System: Open Research Issues, and Solutions, Arabian Journal for Science and Engineering, Vol. 42, No. 2, pp. 399-423, February, 2017.
[122] T. Vissers, T. S. Somasundaram, L. Pieters, K. Govindarajan, and P. Hellinckx, DdoS defense system for web services in a cloud environment, Future Generation Computer Systems, Vol. 37, pp. 37-45, July, 2014.
[123] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, Canada, 2009, pp. 1-6.
[124] B. Gupta and O. P. Badve, Taxonomy of DoS and DdoS attacks and desirable defense mechanism in a cloud computing environment, Neural Computing and Applications, Vol. 28, pp. No. 12, 3655-3682, December, 2017.
[125] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detection, Computers & Security, Vol. 31, No. 3, pp. 357-374, May, 2012.
[126] DARPA KDD 99. (2007). DARPA KDD Cup 1999. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[127] NSL-KDD. (2016). Available: http://www.unb.ca/cic/datasets/nsl.html
[128] CIDD. (2014). Cloud Intrusion Detection Dataset. Available: http://www.di.unipi.it/~hkholidy/projects/cidd/
[129] CAIDA. (2014). Center for Applied Internet and Data Analysis. Available: http://www.caida.org/data/
[130] Shelke, M. P. K., Sontakke, M. S., & Gawande, A. D., Intrusion detection system for cloud computing, International Journal of Scientific & Technology Research, Vol. 1, No. 4, 67-71, 2012.
[131] Hoque, N., Bhuyan, M. H., Baishya, R. C., Bhattacharyya, D. K., & Kalita, J. K., Network attacks: Taxonomy, tools and systems, Journal of Network and Computer Applications, Vol. 40, 307-324, 2014
[132] Cisco, V. (2018). Cisco Visual Networking Index: Forecast and Trends, 2017–2022. White Paper.
[133] Huang, D., Xing, T., & Wu, H. (2013). Mobile cloud computing service models: a user-centric approach. IEEE network, 27(5), 6-11.
[134] K. K. Nguyen, D. T. Hoang, D. Niyato, P. Wang, D. Nguyen and E. Dutkiewicz, "Cyberattack detection in mobile cloud computing: A deep learning approach," 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, 2018, pp. 1-6.
[135] Gai, K., Qiu, M., Tao, L., and Zhu, Y. (2016) Intrusion detection techniques for mobile cloud computing in heterogeneous 5G. Security Comm. Networks, 9: 3049– 3058.
[136] Aldribi A., Traore I., Moa B. (2018) Data Sources and Datasets for Cloud Intrusion Detection Modeling and Evaluation. In: Mishra B., Das H., Dehuri S., Jagadev A. (eds) Cloud Computing for Optimization: Foundations, Applications, and Challenges. Studies in Big Data, vol 39. Springer, Cham
[137] Idhammad, M., Afdel, K. and Belouch, M., 2018. Distributed intrusion detection system for cloud environments based on data mining techniques. Procedia Computer Science, 127, pp.35-41.
[138] Hajimirzaei, B. and Navimipour, N.J., 2019. Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. ICT Express, 5(1), pp.56-59.
[139] Sharma, P., Sengupta, J. and Suri, P.K., 2018. WLI-FCM and artificial neural network based cloud intrusion detection system. International Journal of Advanced Networking and Applications, 10(1), pp.3698-3703.
[140] A. Abusitta, M. Bellaiche and M. Dagenais, "A trust-based game theoretical model for cooperative intrusion detection in multi-cloud environments," 2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), Paris, 2018, pp. 1-8.
[141] C. Lo, C. Huang and J. Ku, "A Cooperative Intrusion Detection System Framework for Cloud Computing Networks," 2010 39th International Conference on Parallel Processing Workshops, San Diego, CA, 2010, pp. 280-284.
[142] A. Abusitta, M. Bellaiche, M. Dagenais, T. Halabi, A deep learning approach for proactive multi-cloud cooperative intrusion detection system, Future Generation Computer Systems, Volume 98, 2019, pp. 308-318.
[143] Gupta, S., Horrow, S., & Sardana, A. (2012). IDS based defense for cloud based mobile infrastructure as a service. Proceedings – 2012 IEEE 8th World Congress on Services, SERVICES 2012, 199–202.
[144] Donald, A. C., Oli, S. A., & Arockiam, L. (2013). Mobile Cloud Security Issues and Challenges: A Perspective. International Journal of Engineering and Innovative Technology, 3(1), 2277–3754.
[145] Kumar, R., & Rajalakshmi, S. (2013). Mobile cloud computing: Standard approach to protecting and securing of mobile cloud ecosystems. Proceedings – 2013 International Conference on Computer Sciences and Applications, CSA 2013, 663–669.
[146] Ali, M., Khan, S. U., & Vasilakos, A. V. (2015). Security in cloud computing: Opportunities and challenges. Information Sciences, 305, 357–383.

[147] Atre, H., Razdan, K., & Sagar, R. K. (2016). A review of mobile cloud computing. Proceedings of the 2016 6th International Conference – Cloud System and Big Data Engineering, Confluence 2016, (July 2018), 199–202.
[148] Abdellaoui, A., Laksantini, A., & Chaoui, H. (2017). A security scheme for mobile cloud using multi-agents system. Colloquium in Information Science and Technology, CIST, 615–620.
[149] Alizadeh, M., Abolfazli, S., Zamani, M., Baharun, S., & Sakurai, K. (2016). Authentication in mobile cloud computing: A survey. J. Network and Computer Applications, 61, 59-80.
[150] Ruay-Shiung-Chang, J. Gao, V. Gruhn, J. He, G. Roussos and W. Tsai, Mobile Cloud Computing Research – Issues, Challenges and Needs, 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering, Redwood City, 2013, pp. 442-453.
[151] H. A. Franke, F. L. Koch, C. O. Rolim, C. B. Westphall and D. O. Balen, Grid-M: Middleware to Integrate Mobile Devices, Sensors and Grid Computing, Proc. 3rd Int'l Conf. Wireless and Mobile Comm. (ICWMC 07), IEEE CS Press, 2007, p. 19.
[152] Ring, M., Wunderlich, S., Gruedl, D., Landes, D., Hotho, A., Creation of Flow-Based Data Sets for Intrusion Detection. In: Journal of Information Warfare (JIW), Vol. 16, Issue 4, pp. 40-53, 2017.
[153] Roesch, M., Snort: Lightweight intrusion detection for networks, Lisa, Vol. 99, No. 1, 1999.
[154] Aceto, G., Ciuonzo, D., Montieri, A., Persico, V. and Pescapè, A., MIRAGE: Mobile-app Traffic Capture and Ground-truth Creation, in Proceedings of 4th IEEE International Conference on Computing Communication and Security (ICCCS 2019), Rome, Italy, October, 2019.
[155] K. Gai, M. Qiu, Z. Ming, H. Zhao, and L. Qiu, Spoofing-Jamming Attack Strategy Using Optimal Power Distributions in Wireless Smart Grid Networks. IEEE Transactions on Smart Grid, vol. 8, no. 5, pp. 2431-2439, Sept. 2017.
[156] K. Gai, K. R. Choo, M. Qiu and L. Zhu, Privacy-Preserving Content-Oriented Wireless Communication in Internet-of-Things. IEEE Internet of Things Journal, vol. 5, no. 4, pp. 3059-3067, Aug. 2018.
[157] K. Gai and M. Qiu, Blend Arithmetic Operations on Tensor-Based Fully Homomorphic Encryption Over Real Numbers. IEEE Transactions on Industrial Informatics, vol. 14, no. 8, pp. 3590-3598, Aug. 2018.
[158] Carol J.Fung, Quanyan Zhub, FACID: A trust-based collaborative decision framework for intrusion detection networks. Ad Hoc Networks, vol. 53, pp. 17-31, Dec. 2016.
[159] Drago, I., et al. Inside dropbox: understanding personal cloud storage services. in Proceedings of the 2012 Internet Measurement Conference. 2012.
[160] Kumar, R., S.P. Lal, and A. Sharma. Detecting denial of service attacks in the cloud. in 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech). 2016. IEEE.
[161] Yu, J., et al., An in-depth analysis on traffic flooding attacks detection and system using data mining techniques. Journal of Systems Architecture, 2013. 59(10): p. 1005-1012.
[162] Koc, L., T.A. Mazzuchi, and S. Sarkani, A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. Expert Systems with Applications, 2012. 39(18): p. 13492-13500.
[163] Peng, J., K.-K.R. Choo, and H. Ashman, Bit-level n-gram based forensic authorship analysis on social media: Identifying individuals from linguistic profiles. Journal of Network and Computer Applications, 2016. 70: p. 171-182.
[164] Rastegari, S., P. Hingston, and C.-P. Lam, Evolving statistical rulesets for network intrusion detection. Applied Soft Computing, 2015. 33: p. 348-359.
[165] Eid, H.F., et al. Linear correlation-based feature selection for network intrusion detection model. in International Conference on Security of Information and Communication Networks. 2013. Springer.
[166] Tsang, C.-H., S. Kwong, and H. Wang. Anomaly intrusion detection using multi-objective genetic fuzzy system and agent-based evolutionary computation framework. in Fifth IEEE International Conference on Data Mining (ICDM'05). 2005. IEEE.
[167] Shafi, K. and H.A. Abbass, An adaptive genetic-based signature learning system for intrusion detection. Expert Systems with Applications, 2009. 36(10): p. 12036-12043.
[168] Mukkamala, S., A.H. Sung, and A. Abraham, Intrusion detection using ensemble of soft computing paradigms, in Intelligent systems design and applications. 2003, Springer. p. 239-248.
[169] Hoque, M.S., et al., An implementation of intrusion detection system using genetic algorithm. arXiv preprint arXiv:1204.1336, 2012.
[170] Shirazi, H.M., An intelligent intrusion detection system using genetic algorithms and features selection. Majlesi Journal of Electrical Engineering, 2010. 4(1).
[171] A. Dainotti, A. Pescape and G. Ventre, Worm Traffic Analysis and Characterization," 2007 IEEE International Conference on Communications, Glasgow, 2007, pp. 1435-1442.
[172] G. Aceto, D. Ciuonzo, A. Montieri and A. Pescapé, "Mobile Encrypted Traffic Classification Using Deep Learning: Experimental Evaluation, Lessons Learned, and Challenges," in IEEE Transactions on Network and Service Management, vol. 16, no. 2, pp. 445-458, June 2019.
[173] Adrienn Dineva, A.M., Sina Faizollahzadeh Ardabili, Istvan Vajda, Shahab Shamshirband, Timon Rabczuk, Kwok-Wing Chau, Review of soft computing models in design and control of rotating electrical machines. Energies, 2019. 12(5).
[174] Salimi, M., et al., State of the art of Machine learning in energy systems. Energies, 2019. 12(5).
[175] Nosratabadi, S., et al. State of the art survey of deep learning and machine learning models for smart cities and urban sustainability. in International Conference on Global Research and Education. 2019. Springer.
[176] Ardabili, S., et al. Deep learning and machine learning in hydrological processes climate change and earth systems a systematic review. in International Conference on Global Research and Education. 2019. Springer.

[177] J. Choi, C. Choi, H. M. Lynn and P. Kim, “Ontology Based APT Attack Behavior Analysis in Cloud Computing,” 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), Krakow, 2015, pp. 375-379.
[178] Marco Balduzzi, Jonas Zaddach, Davide Balzarotti, Engin Kirda, and Sergio Loureiro. 2012. A security analysis of amazon’s elastic compute cloud service. In Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC ’12). Association for Computing Machinery, New York, NY, USA, 1427–1434.
[179] N. Moustafa and J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, 2015, pp. 1-6.
[180] Wu, T., Chen, C., Sun, X. et al. “A Countermeasure to SQL Injection Attack for Cloud Environment.” Wireless Pers Commun 96, 5279–5293 (2017).
[181] Lyu, Y., Mishra, P. A Survey of Side-Channel Attacks on Caches and Countermeasures. J Hardw Syst Secur 2, 33–50 (2018).
[182] Alnaim A., Alwakeel A., and Fernandez E. B., A Misuse Pattern for Compromising VMs via Virtual Machine Escape in NFV. In Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES ’19). Association for Computing Machinery, New York, NY, USA, Article 77, 1–6.
[183] Sharafaldin I., Habibi Lashkari A., and Ghorbani A. A., “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”, 4th International Conference on Information Systems Security and Privacy (ICISSP), Purtogal, January 2018.
[184] Park Y., Zhang Q., Reeves D. and Mulukutla V., "AntiBot: Clustering Common Semantic Patterns for Bot Detection," 2010 IEEE 34th Annual Computer Software and Applications Conference, Seoul, 2010, pp. 262-272.
[185] Aceto, G., Ciuonzo, D., Montieri, A. and Pescapè, A., Multi-classification approaches for classifying mobile app traffic. J. Netw. Comput. Appl. 103, C (February 2018), 131–145.
[186] Persico, V., Montieri, A., and Pescapè, A., On the network performance of amazon S3 cloud-storage service. In 2016 5th IEEE International Conference on Cloud Networking (Cloudnet), Oct. 2016, pp. 113-118.
[187] A. A. Abbasi, A. Abbasi, S. Shamshirband, A. T. Chronopoulos, V. Persico and A. Pescapè, "Software-Defined Cloud Computing: A Systematic Review on Latest Trends and Developments," in IEEE Access, vol. 7, pp. 93294-93314, 2019.
[188] Wang, Maonan, Kangfeng Zheng, Yanqing Yang, and Xiujuan Wang. "An Explainable Machine Learning Framework for Intrusion Detection Systems," IEEE Access (2020).
[189] Otoum, Safa, Burak Kantarci, and Hussein Mouftah. “Empowering reinforcement learning on big sensed data for intrusion detection,” ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, 2019.
[190] A. Dainotti, A. Pescape and G. Ventre, “A Cascade Architecture for DoS Attacks Detection Based on the Wavelet Transform,” Journal of Computer Security, IOS Press, 1 Jan. 2009 : 945 – 968.
[191] Aceto G. and Pescapé A., “Internet Censorship detection: A survey,” Computer Networks, vol. 83, pp. 381-421, ISSN 1389-1286, 2015.
[192] M. D'Arienzo, A. Pescapè, and G. Ventre. “Dynamic service management in heterogeneous networks,” Journal of Network and Systems Management 12.3 (2004): 349-370.