International Journal of Information Management Data Insights 2 (2022) 100134

Contents lists available at ScienceDirect

International Journal of Information Management Data Insights

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

Review

Intrusion detection system in cloud environment: Literature survey & future research directions

Suman Lata [a,1,*], Dheerendra Singh [b]

[a] Research Scholar, College of Engineering and Technology, Chandigarh 160019, India
[b] Department of Computer Science, Chandigarh College of Engineering and Technology, Chandigarh, India

ARTICLE INFO

Keywords:
Cloud security
Intrusion detection system
Cloud datasets
Feature selection
Virtual machine introspection (VMI)
And hypervisor introspection (HVI)

ABSTRACT

The cloud provides infrastructure, applications, and storage services to users that need to be protected by some policies or procedures. Hence, security in the cloud is to protect user data and infrastructure from malicious users by providing Confidentiality, Integrity, Availability, and in-time intrusion detection. The fundamental concept behind the intrusion detection system (IDS) is to identify fraudulent activities to secure user data and cloud services. Therefore, this study provides a coherent view of existing security techniques with their strengths and demerits. It includes security concerns in each cloud service model, the importance of feature selection and dimensionality reduction, and IDS state-of-the-art. This work classifies IDS techniques based on attacks that it identifies, its placement, and configuration. Additionally, the study will also address virtual machine introspection (VMI) and hypervisor introspection (HVI) strategies. The current study is organized on the basis of three distinct perspectives: cloud security concerns, the importance of feature selection, and the analysis of existing IDS techniques. Finally, this work presents a review of existing security issues/challenges and research gaps for future research.

1. Introduction

The cloud computing provides many services to the user namely applications, infrastructure, and storage capabilities. A cloud user can access or manipulate hardware and software according to their needs, principally over the internet. Cloud computing has many benefits for the user but has some limitations and challenges too. The challenges of cloud computing include security, privacy, load balance, cost, and performance management. Out of these challenges, security is the prominent one as user data and applications are on the cloud premises. Cloud computing security comprises policies and procedures to protect cloud-based data, applications and infrastructure from unauthorized access and attacks. It also protects against data leakage, data alteration, software vulnerabilities, SQL injection, cross-site scripting, and flooding attacks (Khalil et al., 2014; Rong et al., 2013). Moreover, cloud subscribers and providers regularly report security issues due to various attacks. For example, in 2012 virtual machine (VM) escape attack was discovered by VUPEN Security (Mimiso, Sept, 2012). Similarly, in 2013, there was a distributed denial of service (DDoS) attack on Dropbox, reported by ENISA (Marinos, 2013), which affected all subscribers by complete service failure for 15 days. And in January 2015, zero day and other more than 450 vulnerabilities were disclosed according to Symantec (2015). In 2018 cloud subscribers faced over 650 million cyber-attacks. Additionally, IoT attacks, distributed denial of service (DDoS) campaigns, targeted ransomware, advanced phishing campaigns, and attacks targeting containers and cloud services were prevalent in 2019.

1.1. Intrusion detection system (IDS)

Furthermore, cloud services are divided into three main categories that are Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). Each of these services and techniques has its own vulnerabilities and issues that must be handled to provide security to the user (Khraisat et al., 2019; Modi et al., 2013). For example, IaaS is vulnerable to attacks on virtual machine images, virtual network assaults, hypervisor attacks, domain name system (DNS) poisoning, ARP or IP spoofing, and cross-site scripting attacks, and data attacks (Aldribi et al., 2020; Jebamalar et al., 1882; Kirat et al., 2014; Prabadevi et al., 2020); PaaS is prone to phishing attacks, Man-in-the-Middle attacks, and port scanning attacks; and SaaS is vulnerable to DoS/DDoS attacks and authentication attacks, and SQL injection attacks as described in publications (Khalil et al., 2014). Therefor there is need of the system that can provide security against attacks and malicious activities. For this reason, intrusion detection system has been developed to provide cloud security. There are two main types of IDS, host-based IDS and network-based IDS. A HIDS monitors

* Corresponding author.
E-mail address: sutharsuman2506@gmail.com (S. Lata).
[1] Present address: Ward No. 21, Udham Singh Chowk, Near S. L. School, Sangaria-335063, Rajasthan (India).

https://doi.org/10.1016/j.jjimei.2022.100134
Received 22 June 2021; Received in revised form 22 November 2021; Accepted 17 October 2022

the operating system, whereas NIDS monitors the network traffic for suspicions activities. A HIDS runs on individual machine to monitor the system calls, important files, and applications to detect internal changes by the insiders. HIDS commonly used to inform the network manager about abnormal behaviour. On the other hand, an NIDS runs on strategic points in the network to monitor internal and external cloud networks. It monitors all devices connected to the network to detect malicious activities or unusual behaviour (Deshpande et al., 2014; Hofmeyr et al., 1998; Patil et al., 2019; Singh et al., 2016). Both techniques have their own advantages and limitations. Hence, combination of these techniques can provide complete cloud security. HIDS secures the machines from insider threats while NIDS monitors device connected to a network such as firewalls, routers, switches, and print servers for outsider attacks. Furthermore, effectiveness of IDS depends on its configuration and technique used for intrusion detection.

1.2. Feature selection

Feature extraction is a technique developed to reduce the dimensionality of the data that can improve the detection accuracy of the system and reduce the false alarm rate (FAR). With the growth of network devices, the volume, variety, and velocity of data from a variety of sources increased at a rapid rate. This data needs to be stored and processed for further use. However, using these data directly as input to the IDS unit is a way to degrade the performance of the system. Because the raw audit data of network traffic is not suitable for intrusion detection because all the features are not helpful to identify an intrusion. Each packet in the network comprises 41 features that generates a number of 2^{41}–1 subsets (Zhang et al., 2020). This huge number of subsets are difficult to handle and also consumes lots of memory and increases the cost. Furthermore, network traffic features are classified into irrelevant features, weakly relevant features, and strongly relevant features depending on their significance in intrusion detection. Hence, to improve the performance and accuracy of the system, the raw date should be pre-processed to reduce dimensionality and to remove the irrelevant features. Feature selection is a technique that can be used to keep the feature that represents original data completely. This technique to removes the irrelevant features and keep the feature that represents original data completely. But it should be done precisely as the accuracy of the system depends on the subset of features selected for intrusion detection. Many researchers have been working to develop feature selection techniques using various algorithms. For example, Khammassi & Krichen (2017) presented a GA based approach for feature selection. According to them, “Deciding upon the right set of features is difficult and time-consuming process”, that can be done by domain knowledge experts. Thus, we need an approach that can automate the feature selection process.

1.2.1. Feature selection techniques

Various feature selection strategies have been introduced. in the literature to improve system performance and reduce memory usage and time. For example, Zhang et al. (2020) proposed an IDS using the feature selection method to improve detection accuracy and efficiency. An article by Prasad et al. (2019) presented an IDS technique using feature selection. They reduced the number of features to 50% of the original set using rough set theory. In their work they showed that feature selection can reduce the system complexity and improve the performance of the system. There are three main methods for feature selection: Filter method, wrapper method, and embedded method.

I Filter technique

This is the most commonly used method for feature selection. In this technique, a threshold value is computed and used to decide whether to keep or discard a feature. The technique is applied directly on the data (Khammassi & Krichen, 2017). It is less expensive as compared to others, but its performance degrades if redundancy is low. Rawashdeh & Al-kasassbeh (2018) used the TShark tool to analyse network traffic and extract useful features for the intrusion detection system.

I Wrapper technique

This method works in three phases, first subsets of features are calculated that represent the data. Then these subsets are classified and evaluated on the basis of some objective function. Lastly, an optimum feature subset is selected to improve the accuracy of the system (Khammassi & Krichen, 2017). A wrapper approach performs better than the filter method, but it requires more computational power and resources. Khammassi & Krichen (2017) proposed a GA-LR wrapper approach for feature selection in network intrusion detection.

I Embedded technique

In this method the system learns the best feature subset while the model is being created. Hence, this method is faster than the filter and wrapper methods. Penalization techniques are used mostly for feature selection in embedded methods, and least absolute shrinkage and selection operator (LASSO) is used for regression. Computation cost for embedded methods is very low, and it is less prone to over-fitting. Patil et al. (2019) developed a NIDS using a feature selection method. In this framework, they have used a binary bat algorithm with two fitness functions for feature reduction. They have reduced the number of traffic features from 44 to 26. After reducing the features, the accuracy of the system was improved. Similarly, Rawashdeh & Al-kasassbeh (2018) and Sakr (2019) used Particle Swarm Optimization (PSO) algorithm for feature selection. Results shows that both the approaches performed better for anomaly detection with increased detection accuracy and reduced FAR. In their work, a comparative analysis is given to show the importance of feature selection.

1.3. Dataset for performance evaluation

For performance analysis of the Intrusion Detection Systems, dataset is required which can represent the real-world scenario of network traffic. Many datasets are used by researchers over the years such as KDD99, NSLKDD, ISC2012 (Citation, 2016; Subhy & Basheer, 2018; Thampi et al., 2019). But these data sets do not reflect the realistic performance of the system, because of missing and redundant records. Record redundancy affects the classifier output as it is partial towards the repeated records. Therefore, there was a need for a dataset that can solve the above two problems. Then Moustafa & Slay (2015) developed a data set in 2015 named UNSW-NB15 to evaluate the performance of IDS. This dataset consists of normal and abnormal traffic records without missing values and redundant records. Each record in this dataset has 47 features generated by matching the output of two tools, namely Argus and Bro-IDS. The output of these tools was stored in a SQL database and matched by using flow features such as: Source/Destination IP address and port number and Transaction Protocol. After that, records are labelled as normal and abnormal records. A normal record was represented by 0 and abnormal by 1. This dataset has many advantages above the previously generated datasets, but it lacks in representing recent attack environments.

To overcome the issues related to UNSW-NB15, Sharafaldin et al. (2018) developed two new datasets, CICIDS2017 and CSE-CIC-IDS-2018. These datasets cover modern day attacks and records and also reflect the current trends. They have used two networks, victim network and attacker network. CICIDS2017and CSE-CIC-IDS-2018 includes six attack profiles: Brute force, heartbleed, botnet, DoS/DDoS, web attack, and infiltration attack. These datasets were created in two steps. First, they have extracted 80 flow-based features from the pcap file. Then they analysed the importance of all the 80 features and the best features are detected using random forest regressor. Selected features are evaluated using machine learning algorithms. To provide a comparative analysis of their work, they compared the proposed datasets with previously available datasets.

The reader can refer to Khraisat et al. (2019) for a detailed description of the characteristics of the data set.

In addition, much research has been done in the field of cloud security to solve security and privacy issues in the last decade. After an initial overview of cloud security issues and solutions, it was found that intrusion detection system is an essential topic to study, as it could protect cloud infrastructure, applications and user data from malicious activities. Therefore, the goal of this work is to review existing IDS techniques, including classification and analysis of existing IDS techniques, as well as their merits and demerits, an overview of various attacks, the importance of feature selection in IDS techniques, and a discussion of available datasets. We will also present research gaps and future research trends for further improvements. The objectives of the present work can be summarized as follows.

- This work aims to analyse existing IDS techniques. Based on type of intrusion detection, its placement and configuration, we have classified IDS techniques in five categories including, (1) Signature-based
- IDS (2) Anomaly- detection-based IDS (3) VM introspection-based IDS (4) Hypervisor Introspection Based IDS and (5) Hybrid IDS technique.
- A secondary goal of this study is to discuss the importance of feature selection. As it improves the accuracy and performance of the intrusion detection system.
- We also summarize security issues and attacks in cloud service models.
- This study also explored current research gaps and future research trends to improve security and privacy.

1.4. Need for cloud security

The cloud provides its services to its users over the Internet, which increases the security risks to both the cloud provider and the user. These are some factors that affect security and privacy in cloud infrastructure.

- The main factor in cloud security is that customer data and programs reside on the provider's premises.
- Cloud providers do not allow the user to implement their security tools that extend into the management layer due to privacy.

  Furthermore, the cloud is based on virtualization, where resources are shared between clients, making it more prone to privacy issues and challenging to develop a security model.

- The primary security challenge faced by cloud providers is from attacks. These can be from the provider side or the subscriber side. Therefore, a system is needed that can provide protection against malicious activities and attacks.

The rest of the paper is organized as: In Section 2 we discuss related work. Section 3 summarizes how we searched for existing techniques and what sources of publications are used. A detailed analysis of existing IDS techniques is given in Section 4. In Section 5, we will provide results and discussion based on different parameters followed by open issues and future research trends in Section 6. Section 7 presents the conclusion of the work and future scope.

2. Related surveys

We studied the existing research literature before proposing our own survey. Several studies have been published that examine the impact of attacks and vulnerabilities in cloud computing, such as Zhou et al. (2010) discussed three basic requirements of cloud security that are confidentiality, availability, and integrity, and Modi et al. (2013) discussed various attacks, vulnerabilities, threats, and security issues at each layer of cloud computing, Denz & Taylor (2013) focused on cloud resiliency, malware, and virtual machine manager (VMM) security, Pandeeswari & Kumar (2015) discussed traditional attacks and how machine learning can handle them, and Khan (2016) provided a threat model for different attacks and their solution. These surveys addressed the issues and factors that affect cloud security, not the solution.

Furthermore, Alhenaki et al. (2019) discussed various attacks in IaaS, PaaS and SaaS. They also provided a detailed specification of threats in cloud computing such as data Loss, data breaches, malicious insider, account and service hijacking. Various security attacks and their solutions are also discussed in this paper. Similarly, Jebamalar et al. (1882) presented eight common reasons that can affect cloud confidentiality, integrity, and availability. This paper also discussed different levels of attacks, the surface of each attack, threats, and vulnerabilities in the cloud environment. The author also discussed the requirement of security in the cloud. But suffers the same limitation, as no discussion about the security technique to tackle these issues in cloud computing. Conversely, Arjunan & Modi (2017) and Azeez et al. (2020) presented literature survey of the intrusion detection system. Both the articles are limited to IDS technique like signature-based IDS, anomaly-based IDS and hybrid techniques without consideration of VM Introspection (VMI) and Hypervisor Introspection (HVI).

After an initial overview of related surveys, we examined that many surveys discussed cloud computing security, cloud attacks, intrusion detection systems, and intrusion prevention systems. However, none of these surveys discussed VMI and HVI techniques of intrusion detection. Another critical constraint on all the work discussed in this area is that no survey included the importance of feature selection and datasets. Feature selection techniques can improve the performance and accuracy of the system and reduce the total cost of security. Therefore, the present work highlights the existing IDS techniques, the importance of feature reduction, and datasets. We will also compare this work with different surveys discussed here on the basis of various parameters. In Section 5.4 we present a comparative analysis of our work.

3. Research methodology

To plan this survey, we followed the systematic review method of Kitchenham et al. (2009), Charband & Navimipour (2016). The main goal of this project is to determine the present state of available IDS approaches, as well as their benefits and drawbacks. Many strategies are used to secure cloud infrastructure, which must be evaluated on the basis of various security requirements. In this paper, we offer and explore a variety of techniques for providing security at various levels of cloud architecture. To conduct this study, we searched for articles about cloud security in reputable journals, conferences, and publications from January 2010 to June 2020. This includes Springer, ScienceDirect, Scopus, IEEE Xplore, ACM Digital Library, and Google Scholar.

Furthermore, we have defined some relevant keywords to search articles in the aforementioned databases. These keywords are related to security challenges and the solutions that have been taken to address them, specifically in the cloud context. The following are some of the most commonly used keywords: "security", "intrusion detection," "intrusion prevention," "feature selection," "importance of dimensionality reduction", "attacks," "security challenges," and “datasets” "in cloud computing." After reading the abstracts, we decided which articles were relevant to our work and which were not. Selected publications are examined and analysed to provide a survey of existing IDSs in cloud computing for dealing with various intrusions. The percentage of information sources used in this study is shown in Fig. 1.

4. Survey of existing IDS cloud computing

This section classifies and analyses various intrusion detection techniques. In this work, we have classified IDS techniques based on their configuration, placement, and attacks it detects into five categories as

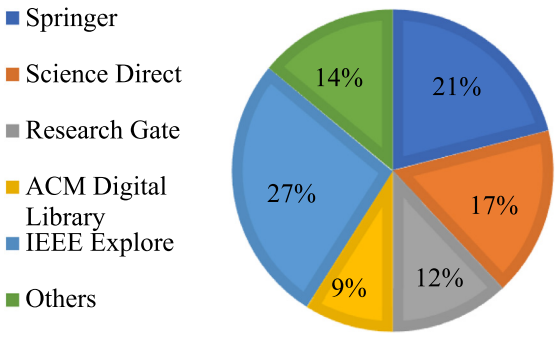

Fig. 1. Source of research articles surveyed included in present work.

shown in Fig. 2. The following subsections discusses these techniques in detail. And, a summary of the existing IDS technique with its characteristics and limitations is given in Tables 1–5.

4.1. *Signature-based IDS*

Signature-based IDS detects current suspicious activities by comparing it with known patterns or malicious instructions. It maintains a signature database to keep a record of various attacks and malicious activities. This signature database needs to be updated regularly to detect recent attacks. To detect a malicious activity, current network packets are compared with the stored set of rules. One of the simplest and widely used signature-based methods is Snort, presented by Martin Roesch (2015). Snort is a very popular signature-based IDS for packet capturing and real-time network traffic monitoring. Fig. 3, shows its major components that are packet decoder, pre-processor, detection engine, logging and alerting system. Current traffic is first pre-processed and then transferred to the detection engine. Pre-processing eliminates redundant and incomplete data. The detection engine then compares the current packet with records stored in the signature database. If there is a match, then the alarm is generated for the concerned authority; otherwise, the packet is passed as a normal packet (Roesch, 2015).

Many researchers presented signature-based IDS such as Lin et al. (2012) proposed a rule-based NIDS to detect known attacks in a cloud environment. To configure the detection rules, information from the operating system of each VM is collected and updated dynamically. Lo et al. (2010) presented a cooperative intrusion detection framework (CGA). Each server has an IDS that is a combination of a signature database and a block table that maintains the record of recent attacks. Snort compares the packet to the block table first, then to know signatures. Because the likelihood of recent attacks is higher, they should be checked first. Information about abnormal packet is transfer alert clustering, which uses a threshold value to judge the severity of the packet. A malicious packet is then dropped by the IDS unit. Similarly, Meng et al. (2014) proposed signature-based IDS techniques. According to the authors, in a malicious network traffic situation, the probability of a mismatch is greater than the probability of a match. Hence, they used a mismatch policy to identify the attack. Mandal et al. (2015) proposed signature-based IDS technique to detect application-level attacks. In this technique, a sniffer is placed between the cloud provider and the user, which captures the packets and transfers them to the parser for further processing. A parsing grammar analyses the parser output against stored semantic rules, and the result is generated accordingly.

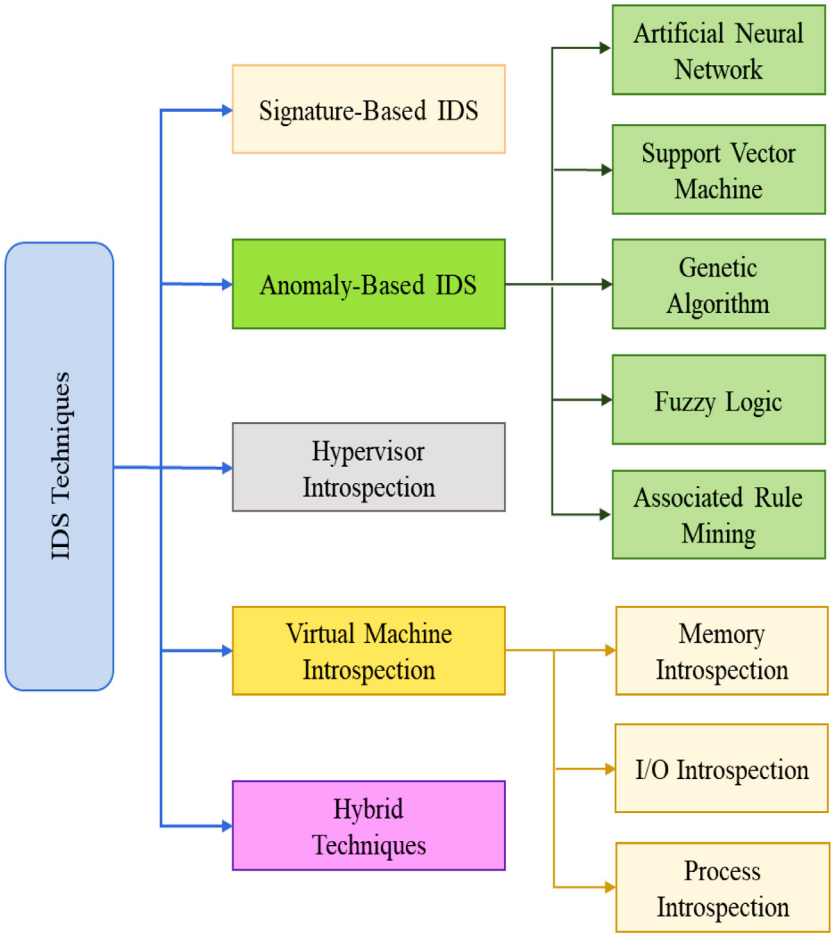

Fig. 2. Classification of IDS techniques.

Table 1
Summary of existing signature-based IDS techniques.

Refs.	Attack Detected	Dataset	Characteristics	Limitation
(Lo et al., 2010)	Network attacks (VM)	Simulation environment	- Uses Snort to detect known attacks. - The detection unit placed over each VM.	- Vulnerable to a zero-day attack - Need to maintain a database. - Increased system overhead due to number of IDS units.
(Lin et al., 2012)	Network attacks (VM/ VMM)	Simulation environment	- NIDS is placed in VMM's privileged domain. - Knowledge-based approach is used.	- Cannot detect novel attacks. - Vulnerable to viruses and worms.
(Meng et al., 2014)	Signature-based attacks	Simulation environment	- Based on the mismatch technique. - Host and cloud data analysed parallelly to reduce the time consumption.	- Low performance as two rounds are required to detect intrusion.
(Mandal et al., 2015)	Network attacks	Not mentioned	- Based on semantic rules. - Parsing grammar identifies the intrusion by checking the output of parser against semantic rules.	- Needs to maintain a database. - Vulnerable to a zero-day attack
(Aldwairi, 2017)	Network anomalies	New cloud intrusion dataset	- Reduced execution time and memory usage.	- Cannot detect novel attacks
(Santoso et al., 2016)	Network attacks	Simulation environment	- Uses Snort for known attacks. - Detects UDP flood DoS attack successfully in an open stacks private cloud.	- Only detects UDP flood attacks. - Vulnerable to other DoS attacks, such as TCP SYN and EDoS.

Table 2
Summary of existing anomaly detection IDS techniques.

Refs.	Attacks	Dataset	Characteristics	Limitation
(Kumar et al., 2011)	Attacks on data	Not specified	- Uses hidden Markov technique to produce a behavioural model. - Low storage requirement.	- Increases vulnerabilities, if its losses order of system calls - Increased false positives
(Yuxin et al., 2011)	Network traffic attacks	Simulation environment	- Identifies malware code installed in VM. - Based on machine learning algorithms.	- Vulnerable to Code obfuscation techniques. - On every program change, CGG maintenance is required.
(Srinivasan et al., 2012)	Network attacks (VM)	Not specified	- Detects IP Spoofing DDoS and port scanning attacks.	- Increased False Alarms.
(Wolthusen, 2012)	Malware attacks (VM)	Linux KVM-based reference scenario.	- Maintains a log file of each system call. - Low storage requirement	- Increase in vulnerabilities, if losses order of system calls. - Increased false positives
(Deshpande et al., 2014)	Malicious activities in the system	CICIDS 2017 and CSECICIDS 2018 (Sharafaldin et al., 2018)	- Less computational overhead. - Improved detection sensitivity.	- This method analyses - selective and failed system calls only within the system.
(Gupta & Kumar, 2015)	Malware attacks (VM)	UNM (University of New Mexico) (Hofmeyr et al., 1998)	- Detect malware attacks by analysing system calls.	- Requires more storage. - Increased false positive
(Pandeeswari & Kumar, 2015)	Network attacks (VM/VMM)	DARPA KDD 1999 (Citation, 2016)	- Reduces the false alarm rate. - Improves detection accuracy.	- Does not evaluated against recent attacks.
(Zhang et al., 2020)	Network anomalies	NSL-KDD (Subhy & Basheer, 2018)	- Reduces the false alarm rate. - Feature Optimization was used to increase Recall and Precision.	- Reduces the false alarm rate. - Performance of the system was affected by the increased number of parameters. Hence, there is a need for classifier improvement.
(Intelligence et al., 2020)	DoS/DDoS, botnet, brute force, port scan and web attacks	tCICIDS 2017 and CSE-CICIDS 2018 (Sharafaldin et al., 2018)	- Achieved 99% accuracy with 0.5% FAR - Can detect 0-Day Attacks.	- Accuracy and false-positive rate of the detection system depends on the number of hidden ANN layers.
(Pacheco et al., 2020)	Cyber and flooding attacks and anomalies due to misuses, or system glitches	Simulation environment	- A Riemann rolling feature extraction scheme was used to improve performance. - A new dataset was introduced which includes multistage attacks.	- Meltdown, spectre and VMescape attacks were not considered.
(Rawashdeh & Al-kasassbeh, 2018)	DDoS attack	Simulation environment	- ANN was used to reduce false alarm. - Increased detection accuracy.	- Can only detect two types of attacks: UDP flood and TCP Syn. - Only analyses VM to VM traffic. Hence, vulnerable to outside attacks.
(Sakr, 2019)	Network attacks	NSL-KDD (Subhy & Basheer, 2018)	- Achieved higher true positive rate (TPR), true negative rate (TNR), and low false positive rate (FPR) with increase classification accuracy.	- Cannot guard against VM escape inter VM attacks.
(Prasad et al., 2019)	DoS/DDoS, Heartbleed, web attacks, and port scan	Simulation environment	- Reduced time and Space complexity.	- Range of estimated probability for feature selection and pre-processing steps were done manually.

Table 3
Summary of existing VMI IDS techniques.

Refs.	Attacks	Dataset	Characteristics	Limitation
(Maiero & Miculan, 2011)	Decode, Syslog, and Forwarding loop	UNM (University of New Mexico) (Hofmeyr et al., 1998)	- General interrupts from CPU registers were used to collect VM information for intrusion detection.	- Interrupts tracking increases system overhead. - Requires special expertise for information interpretation.
(Benninger, Neville, Yazir, Matthews & Coady, 2012)	Malware attacks (Rootkit) (VM/ VMM)	Simulation environment	- A threshold value is used to classify hyper-calls.	- Restricted to para-virtualized systems. - The author did not explain the technical details.
(Lengyel et al., 2014)	Decode, Syslog, SScp	Simulation environment	- Memory introspection technique used to access the file system. - Can monitor suspicious drivers and rootkits.	- Kernel's expertise is required. - Slows down the system.
(Shi et al., 2016)	Hypercall-based attacks (VM/VMM)	Simulation environment	It can trap both user-level and kernel-level functions. Detects hypercall-based attacks.	- Doesn't mentioned the technique used to filter the kernel and user functions. - Cannot detect system call-based attacks.
Kumara and Jaidhar, 2015 (AKMA, 2016)	Suspicious activities in guest OS	Simulation environment	Detects average rootkit and jynx rootkit successfully.	- Hypervisor is not secured.
(Borisaniya & Patel, 2019)	Malware Such as worms and trojan	Simulation environment	- This technique can monitor multiple process on multiple VMs, hosted on multiple physical machines.	- There is an increase in the average response time with the increase in the number of VMs or Host machine.
(Mishra et al., 2019)	Malware (Subversion attacks)	Malware Sample of windows binaries collected from University of California for experimental setup (Kirat et al., 2014)	- Can detect malware (diamorphine) and attacks like conficker and torpig	- The security tool was deployed on hypervisor that can be compromised.
(Jia et al., 2017)	Malware (Subversion attacks)	Simulation environment using ARM Foundations Model 8.0	- Proposed T-VMI, where the monitoring system is secured.	- Needs a hypervisor modification. - 5% performance loss.

Santoso et al. (2016) proposed a Network Intrusion Detection System using SNORT for the open stack cloud. The author designed a NIDS to classify various attacks and concluded that a Denial of Service (DoS) attacks are possible by the user datagram protocol (UDP) flood. Open stack private cloud used for performance evaluation of the proposed system. Aldwairi (2017) presented a signature-based IDS using the Myer algorithm for the MapReduce framework. They used a multi-core CPU to parallelize the signature matching operation and reduced execution time and memory usage. A summary of existing signature-based IDS techniques is presented in Table 1.

4.2. Anomaly-detection-based IDS

Although a signature-based IDS detects known attacks at high speed and has very low false positive rate, but it requires regular maintenance of signature database. To overcome the limitations of signature-based IDS, anomaly detection techniques were developed. This technique analyses the user behaviour to create a behavioural profile. Then, this behavioural profile is used to identify both known and unknown attacks. Fig. 4 shows a basic working model of an anomaly detection technique. It includes two main phases, namely, the training phase and the detection phase. In training phase, feature construction module collects data from host machine or network and pre-processes it to construct features. These features are used by training module to generate a behavioural model. This model categorizes the data as normal or abnormal (intrusion) behaviour. The anomaly detection phase uses this model to detect intrusion. Any deviation from normal traffic is considered as intrusion and an alarm is generated to the security administrator (Sari, 2015). This technique can detect novel attacks, but requires more computational power. As any deviation from normal behaviour generates alarm, now it is up to the security manager to identify the reason for the alarm.

Anomaly based techniques are further classified based on the technique used to detect anomalies such as Machine Learning, Fuzzy logic, Support Vector Machine, and Data Mining. In the last decade, several studies have explored these techniques. For example, Kumar et al. (2011) used a hidden Markov technique to produce the behavioural model. This technique uses a log file of system call frequencies to detect malicious activities. An IDS is developed using three profiles: low, middle, and high, each of which corresponds to the features of recent activity. Patterns having a low probability of matching are represented by the low profile. The high profile, on the other hand, relates to patterns with a very high chance of matching. Finally, the middle profile is the present profile that matches partially. Every profile is matched based on a predetermined threshold value.

Yuxin (2011) proposed an approach using machine-learning technique based on static program behaviour analysis. It works in two phases: the first step is to decode programs and then to create context-free grammar to represent the process flow. To obtain the complete sequences, we explore and assembled all of the branches. All system calls are reduced to a few short sequences. They employed two different feature selection techniques: Information Gain (IG) and Document Frequency (DF) separately. Srinivasan et al. (2012) proposed an IDS technique using two-tier system which is a combination of unsupervised learning and supervised classification. Wolthusen (2012) used frequencies of normal/abnormal system calls for intrusion detection. Firstly, a huge number of records are collected from each VM over a period of time. The methos presumes that the VMs are not malicious for a period of time after initialization. It has the time complexity O (n), where n is the total number of lines. This technique has a 100% detection rate with 11% false positives.

SyedNavaz et al. (2013) proposed an entropy-based IDS to detect unknown attacks in the cloud environment. To detect low-frequency attacks Gupta & Kumar (2015) devised a system call-based anomaly detection approach. Rather than employing a training system, this method generates a database of system calls that is designed by a pair of keys. During execution, one key represents the name, and the other indicates the immediate successor. The activities to be tracked are determined by comparing them to the baseline database, which is established by

Table 4
Summary of existing HVI IDS techniques.

Refs.	Attacks	Dataset Used	Characteristics	Limitation
(Zhang et al., 2012)	Data Leakage (VMM/VM)	Simulation environment	- To separate the security functions from VMM a small layer is added below hypervisor, to secure the guest VM data. - This technique does not require any VMM modification.	- Slows down the system. - Extra layer increases the attack surface.
(Wang et al., 2012)	Memory attacks and misuse of HyperLock Services	Simulation environment	- Hypervisor runs in separate address space. Therefore, attackers cannot attack other virtual machines in the cloud from the compromised hypervisor.	- Reduced System performance. - Hypervisor design requires huge modification - It cannot detect side-channel attacks.
(Ding et al., 2013)	Control data attacks	Simulation environment	- It can analyse scheduler data, security policy data, and privileged data.	- Cannot guard against VM Escape inter-VM attacks. - Performance degradation.
(Wang et al., 2010)	Rootkit targeting the integrity of OS & hypervisors	Simulation environment	- Uses a stored image of memory and CPU register to check the integrity of the system.	- It can detect hardware attacks but failed to detect transient attacks. - An alteration of the hypervisor is required. - Increased hardware dependency.

Table 5
Summary of existing hybrid IDS techniques.

Refs.	Attacks	Dataset used	Characteristics	Limitation
(Ficco et al., 2016)	Distributed attacks (DoS/DDoS)	Simulation environment	- Based on Security as a Service principal model.	- Require an increase of no of security tools as probes are installed on each VM.
(A Collaborative Intrusion, 2022)	Distributed attacks	No performance evaluation done.	- Detects distributed attacks. - Fast detection rate. - Reduced FAR.	- A security tool was deployed at each VM that increases the computational cost.
(Chiba et al., 2016)	Internal & external attacks from Physical and Virtual Network	No performance evaluation done	- Uses Optimization techniques to improve the system performance.	- Increased communication overhead. - Not evaluated against recent attacks.
(Al Haddad et al., 2016)	Network anomalies & distributed attacks	Simulation environment	- The detection component is placed on a hypervisor to detect coordinated attacks.	- Increased communication overhead due to alert generation. - Not evaluated against recent attacks.
(Singh et al., 2016)	Distributed attacks	KDD99 (Citation, 2016), NSL-KDD (Subhy & Basheer, 2018)	- NIDS is installed in a network bridge using a centralized approach. - Less communication overhead.	- Less scalable as compared to distributed approaches.
(Balamurugan & Saravanan, 2019)	DDOS, U2R, 0-day, flood, and R2L attacks	UNSW-NB15 (Moustafa & Slay, 2015) And CICIDS-2017 (Sharafaldin et al., 2018)	- Uses the cloud controller, trust authority, and VMM.	- Cannot detect VM-to-VM attacks. - Can only work for a specific server. Hence, not suitable for multiple servers.
(Arjunan & Modi, 2017)	Virtual network attacks such as buffer overflow & distributed attacks	Offline simulation using different intrusion datasets	- IDS is deployed at a virtual network to monitor each VMs and can also detect physical network attacks. - Also capable of detecting distributed attacks.	- To handle high traffic to VM it requires high computational power that increases the cost. - Multiple security tools need to be protected from vulnerabilities.
(Mishra et al., 2017)	Spoofing and virtual network attacks from VM	UNSW-NB (Moustafa & Slay, 2015)	- Uses two levels of security checks that improve the system robustness.	- Specially developed for a particular server. Hence, cannot detect distributed attacks. - To handle high traffic to VM, it requires high computational power - that increases the cost.
(Jung & Zarrabi, 2017)	Internal and external attacks	Eucalyptus installed to create real word scenario	- Used learning vector quantization algorithm for clustering and then decision tree classifier for intrusion detection in each cluster.	- A decision tree is used that has an overfitting problem. - The existence of noise affects the accuracy of the system.
(Patil, 2018)	DoS & DDoS attacks	KDD'99 (Citation, 2016)	- Detects DoS/ DDoS attacks successfully.	- Only detect DoS/DDoS attacks. - Not evaluated against recent attacks.
(Patil et al., 2019)	Virtual network attacks	UNSW-NB15 (Moustafa & Slay, 2015) & CICIDS-2017 (Sharafaldin et al., 2018)	- The proposed system is deployed at the server level. Hence, a compromised VM cannot affect the security system.	- Only newly joined VM was under consideration - VM to VM traffic can't be examined.
(Ahram et al., 2020)	DoS & DDoS attacks	Simulation environment	- GA was used to detect and prevent DoS/ DDoS attacks.	- Increased overhead due to more NIDS units.

the cloud administrator. The anomaly sequence is represented by any mismatch. With a 98% accuracy rate, this approach can detect intruders. This technique can detect intrusions with a 98 percent accuracy rate. Al Haddad et al. (2016) used the support vector machine (SVM) for anomaly detection. Pacheco et al. (2020) developed an IDS based on ANN that detects intrusion at all nodes and then generates an alert. These techniques are limited to detect traffic attacks such as DoS/DDoS, and IP spoofing without considering worms, viruses, and rootkits.

Pandeeswari & Kumar (2015) combined fuzzy C-means and artificial neural networks (ANN) to reduce false alarms and improve the accuracy of the system. The large database is separated into groups using this technique, which will be used to train the various ANN modules.

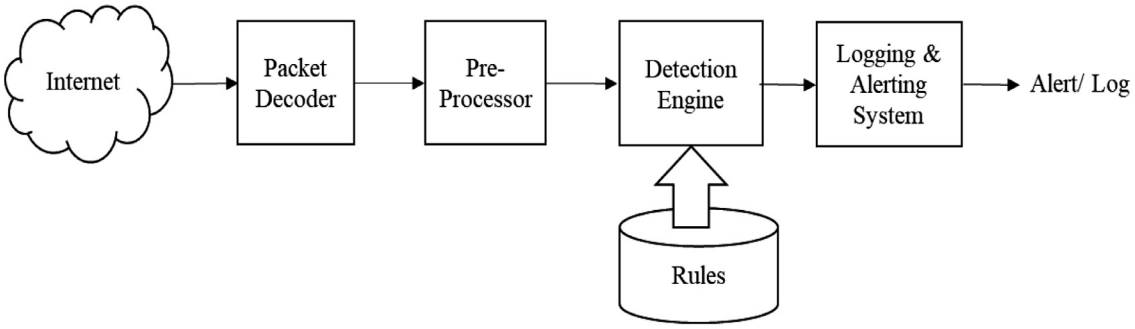

Fig. 3. Signature-based intrusion detection system (Roesch, 2015).

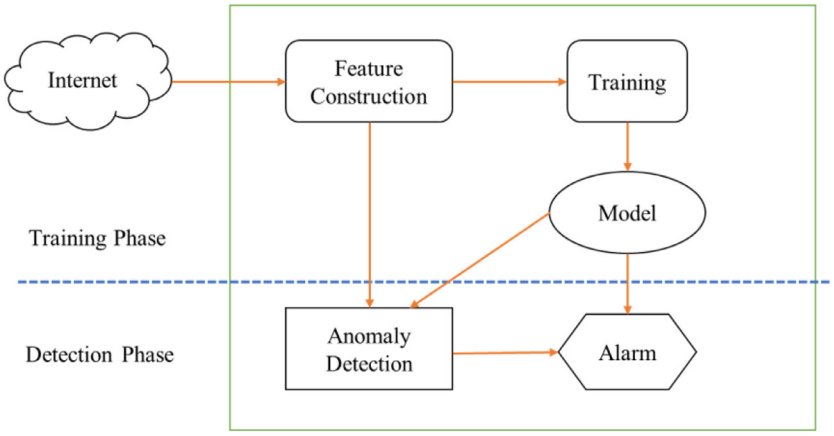

Fig. 4. Anomaly detection model (Pacheco et al., 2020).

The fuzzy segment is then used to combine the results of many ANNs. The results reveal that this technique can detect a wide range of hypervisor assaults with high detection accuracy and a small percentage of false alarms. Similarly, Rawashdeh & Al-kasassbeh (2018) used a combination of PSO and ANN. Where PSO selects appropriate weight for neural network to improve the accuracy. Zhang et al. (2020) used feature optimization techniques to improve system accuracy with the integrated dominance algorithm (MaOEAABC). Deshpande et al. (2014) presented host-based IDS (HIDS) based on the assumption that malicious behaviour is evidently different from normal behaviour. This technique monitored failed system calls to detect intrusion using k-NN. Catillo et al. (Intelligence et al., 2020) used a two-level approach to classify attacks using ANN. In this technique the ANN has three layers, where the I/O layers have the same dimension, but the hidden layer is often smaller than the input. During the training process, the hidden layer catches important features of training data. This technique adjusts the objectives and decision-making principles based on experience to attain 99 percent accuracy with 0.5 percent FAR at the second level. Aldribi et al. (2020) proposed an anomaly detection technique for intrusion detection and an instance-orientated feature extraction technique. Table 2 presents the characteristics and limitations of these techniques.

4.3. Virtual machine introspection (VMI)-based IDS

The placement of IDS placement plays a vital role in the intrusion detection process. If an IDS is placed on host; it has a complete view of the system but is itself prone to attacks. However, if an IDS placed in a cloud network, it is less prone to attacks but there is a reduced system visibility. Similarly, if an analysis is performed on the tenant VM, then the analysis component can be compromised by advanced malware programs. And, it becomes easy to breach the monitored VM security. Therefore, a technique that can isolate IDS from the monitored VM is needed. It can be done by deploying a security tool in Virtual Machine Manager (VMM). This intrusion detection technique is known as Virtual Machine Introspection (VMI), where VMs are monitored from outside by collecting data at hypervisor level (Pfoh et al., 2022) A basic model of the VMI technique is shown in Fig. 5, where the monitoring VM extracts state information from the monitored VM using the Lib VMI library. The Lib VMI library is the core component of the VMI technique and is used to interpret information in VM from raw data (Borisaniya & Patel, 2019). Many methods used by researchers to collect data from monitored VM such as memory introspection, system events or process introspection, and Input/Output (I/O) introspection. In memory introspection-based approaches, data are collected from main memory. I/O introspection deals with hardware communications such as file system call introspection, interrupt request introspection, and system call (a service request to the kernel) introspection (More & Tapaswi, 2014).

Many researchers used the VMI technique for intrusion detection, such as Bharadwaja et al. (2011) proposed a VMI based approach based on hyper call authentication, namely 'Collabra'. This strategy works as a filtering mechanism for guest VM initiated hyper-calls because guest VMs are exposed to the outside world and are inherently susceptible. The goal is to keep the VMM safe while also ensuring that other guest VMs receive uninterrupted service. This method also separates the damaged module from other VMMs, preventing it from communicating with them. But this technique increases the traffic overhead. Maiero & Miculan (2011) proposed a process introspection technique to monitor the target machines' processor context and traps the general-purpose interrupts. Lengyel et al. (2014) developed a system based on a kernel debugging approach to monitor the kernel file system. The author used LibVMI for direct memory access. AKMA (2016) used the system calls in detail and opened known backdoor ports to detect malicious activities. Borisaniya & Patel (2019) proposed a two-phase VMI-based approach to monitor user in-VM activity. First, the author used system calls to generate training data during the learning phase. Then, these data of feature vectors are used for classification purpose in detection phase.

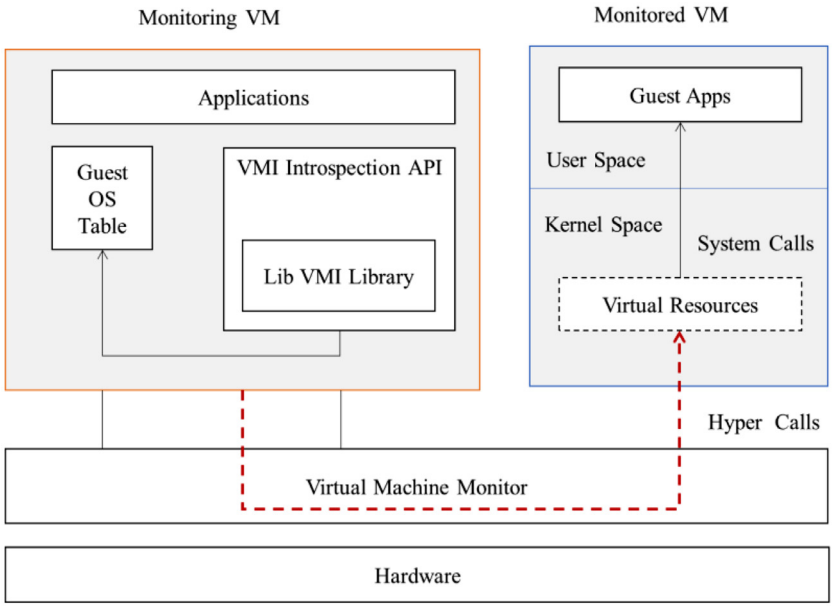

Fig. 5. Virtual machine introspection (AKMA, 2016).

Laurén (2018) introduced a Python-based Nitro VMI framework and how it can be used for application development. In this work, Nitro web is used to monitor VMs for intrusion detection. A VMI technique is developed on the assumption that a hypervisor is a secure place to deploy an IDS. But this assumption can be violated by insiders in cloud environment. Therefore, Jia et al. (2017) proposed a T-VMI (Trusted-VMI) technique that can protect the detection unit from malicious subversion. Mishra et al. (2019) proposed a VMI technique to detect rootkit attack and malware. They accessed CPU register and memory for low level information for intrusion detection. Table 3, presents characteristics and limitations of these technique.

4.4. Hypervisor-introspection (HVI)-based IDS

Although in VMI techniques monitoring agent is isolated from monitored machine but for dynamic analysis of the program, it is required to place an agent in the running environment, and this can be compromised by an attacker. Additionally, the VMI technique works under the assumption that the hypervisor is a secure place to run the VMI tool. However, a report published by NIST in 2014 states that the Xen hypervisor and VMWare ESX hypervisor are vulnerable to attacks. A compromised hypervisor can be used to launch attacks on VMs. So, a new technique named Hypervisor Introspection (HVI) was proposed that does not deploy security tool at hypervisor but below the hypervisor level. This technique examines the control and noncontrol flow data, memory, hypercalls, and data structure related to the hypervisor. Many researchers used this technique in literature for intrusion detection. Such as Zhang et al. (2012) proposed a nested virtualization-based security approach where a security model is deployed below the Virtual Machine Monitor (VMM). It is based on the assumption that if VMM gets compromised; it does not affect the guest VM data. Similarly, Wang et al. (2012) proposed the HyperLock technique, which uses a sperate address space to run the hypervisor. According to this work, other virtual machines in the cloud are safe in case a hypervisor is compromised by an attacker. Ding et al. (2013) proposed Hyper-Verify, that can analyse scheduler data, security policy data and, privileged data. Hyper-Verify improves the flexibility of the system because it does not depend on the hardware of the system, and an IDS can be improved even after its deployment.

Furthermore, Wang et al. (2010) proposed Hyper-Check, which works in three modules. Where, Memory acquisition and CPU register checking modules run on the monitored machine and the third one, which is an analysis module deployed on a monitoring machine. The gained memory image is compared with stored memory image to verify the integrity of data, likewise CPU register data integrity is verified. The report generated by these modules is sent to the monitoring machine, and an alert is generated accordingly. This technique is known as a Hypervisor-Assisted integrity framework. Shi et al. (2016) proposed a system that can analyse the running program in VMs by using VMM Introspection as shown in Fig. 6. Table 4, presents characteristics and limitations of these technique.

4.5. Hybrid technique

Many techniques have been developed in the literature to address security issues in the cloud environment. As discussed in previous subsections signature-based IDS are fast and simple, an anomaly-based detection technique detects novel attacks, VMI technique isolates the security tool from target machine, and HVI technique can detect hardware attacks. But each technique lacks some capabilities that the other contains. Additionally, a single IDS technique could not detect all attacks in a cloud environment. Thus, a proper combination of different IDS techniques creates a truly robust defensive network. And this combination is known as a hybrid intrusion detection system. Many researchers presented various combinations of IDS techniques for intrusion detection in cloud environment. Some of these are, Ficco et al. (2016) presents a hierarchical architecture of multiple security components to provide security as a service. They installed probes in every virtual machine to collect the information. The collected information is then passed to an upper-level security engine, which decides whether the data are normal or an attack. Chiba et al. (2016) used a combination of Snort (a signature-based IDS) to detect known attacks and Back Propagation Neural Network for anomaly detection. They deployed NIDS on each processing server that works cooperatively to provide security against insider and outsider attacks. An alert system collects information from all detection units and generates an alarm accordingly. Al Haddad et al. (2016) combined Snort and SVM (Support Vector Machine) for anomaly detection. Singh et al. (2016) used a combination of the decision tree classifier and SVM for anomaly detection and Snort for known attacks. Balamurugan & Saravanan (2019) used two algorithms, the packet scrutinization algorithm and the clustering algorithm K-means with ANN to analyse the network traffic of users. Arjunan & Modi (2017) used Snort to detect

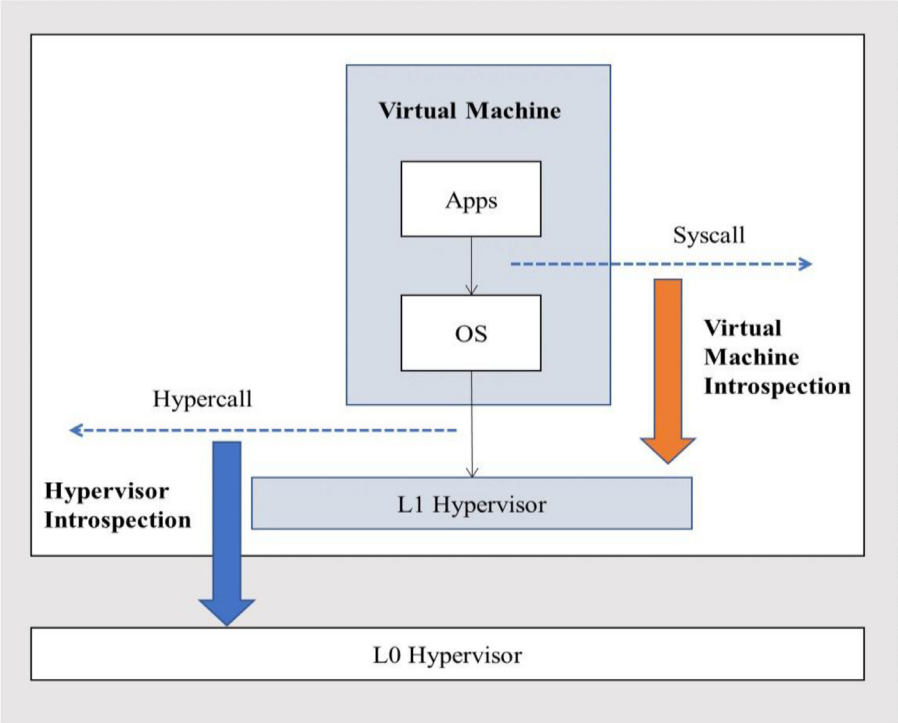

Fig. 6. Hypervisor introspection (Shi et al., 2016).

known attacks and then various classifiers such as Naive Bayes, Decision Tree, Random Forest, and Linear Discriminant Analysis for anomaly detection. For the final results, a decision-making unit then uses the Dempster-Shafer Theory (DST) to the results collected from various classifiers Mishra et al. (2017) proposed an approach in which the Security tool was deployed on the cloud network server, which can inspect the VM-to-VM and VM-to-an outside cloud network and vice versa. Jung & Zarrabi (2017) used Snort for known attacks and Learning Vector Quantization Algorithm for clustering.

Patil (2018) used Snort for known attacks and then features are extracted from each packet after that classification is done using three algorithms Decision tree, Random Forest Classifier and OneR. Patil et al. (2019) presented the hybrid HLDNS method, which is placed on each physical server's Control VM (CVM). It maintains a track of virtual network layer traffic as well as traffic from external networks. When a new VM is added, the proposed framework uses the Libcap library to start capturing its network traffic. Then, to detect known attacks, Snort is employed. The packets are classified into normal and abnormal using a random forest classifier. After collecting information from snort and anomaly detection, the alert generating module generates an alert. In addition, the binary bat Algorithm is used for feature selection to optimize the results. Furthermore, Prasad et al. (2019) presented an IDS using the rough set theory and the Bayes Theorem. In this approach, first important features are selected based on probability to reduce the system complexity. The dataset was then classified using Bayesian set theory into three categories: normal (belongs to the target class), abnormal (does not belong to the target class), and intermediary (may belong to the target class). Ahram et al. (2020) proposed a hybrid approach using Snort and Genetic Algorithm (GA) to provide Availability, Confidentiality and Integrity to the cloud resources and services. They improved Snort rules to detect DoS/ DDoS attacks and used GA for anomaly detection. NIDS is placed in each cloud cluster and on each link joining these clusters. A Collaborative Intrusion, 2022) proposed a host-based intrusion detection and prevention approach using signature matching and anomaly detection techniques. The security tool was deployed at each VM, to monitor the activities of machine. Table 5, presents characteristics and limitations of these technique.

4.6. Challenges faced by each IDS

So far, we've looked at some of the existing techniques of IDS into the cloud. However, no universally effective method has yet been discovered. Each has its own set of constraints. Table 6 summarizes the challenges faced by each approach.

5. Result and discussion

In Section 4 we have discussed and classified existing intrusion detection techniques used for cloud security. For future research work, in this section we present the results attained from the comparative analysis of existing intrusion detection technique based on different criteria:

5.1. Based on technique used

Many techniques are presented in the literature for intrusion detection in cloud computing. We have classified them into five categories, as mentioned in Section 4. Here we present a year-wise summary of different techniques; this would be useful in better understanding which techniques were primarily used in earlier years and how new techniques were further introduced. Here, we split the surveyed techniques into three time periods, 2010–2013, 2014–2016, and 2017–2020. As can be seen in Table 7, fourteen articles were selected during the 2010–2013 period. Similarly, twelve articles were selected during the 2014–2016 period, and seventeen articles from 20,172,020.

Furthermore, out of 43 techniques discussed in this work, the percentage of Signature-Based, Anomaly Detection-based, VM Introspection, Hypervisor Introspection, and Hybrid techniques is 14%, 30%, 19%, 9%, and 28% respectively. Fig. 7 presents the percentage of each technique during time period 2010–2013, 2014–2016 and 2017–2020.

5.2. Based on feature selection

In this work, we have discussed the importance of feature selection to improve system performance. Only 6 articles; Patil et al. (2019), Zhang et al. (2020), Prasad et al. (2019), Rawashdeh &

Table 6
Challenges faced by each IDS.

IDS	Challenges
Misuse Detection	- It is not possible to detect novel or variants of known attacks. - Need to maintain signature dataset. And it is difficult-to-keep-up-to date signature dataset for matching. - It generated a high false alert rate (FAR) for unknown attacks.
Anomaly Detection	- It takes longer to identify attacks. - The accuracy of detection is determined by the amount of data and its properties.
VMI	- Require special expertise. - The IDS running environment can be compromised. - Hypervisor is not secured.
HVI	- Because this strategy is rarely employed, it can be difficult to understand. - Lack of experience. - Cannot detect hardware attacks.
Hybrid	- Increases processing overhead for detecting intrusions.

Table 7
Classification of articles based on technique used.

Technique	2010–2013	2014–2016	2017–2020
Signature Based IDS	(Lin et al., 2012; Lo et al., 2010; Meng et al., 2014)	(Mandal et al., 2015; Santoso et al., 2016)	(Aldwairi, 2017)
Anomaly Detection IDS	(Kumar et al., 2011; Srinivasan et al., 2012; Wolthusen, 2012; Yuxin et al., 2011)	(Deshpande et al., 2014; Pandeeswari & Kumar, 2015; Gupta & Kumar, 2015)	(Intelligence et al., 2020; Pacheco et al., 2020; Prasad et al., 2019; Rawashdeh & Al-kasassbeh, 2018; Sakr, 2019; Zhang et al., 2020)
VMI Based IDS	(Benninger et al., 2012; Maiero & Miculan, 2011)	(AKMA, 2016; Lengyel et al., 2014; Shi et al., 2016)	(Benninger et al., 2012; Jia et al., 2017; Mishra et al., 2019)
HVI Based IDS	(Ding et al., 2013; Wang et al., 2012; Zhang et al., 2012)	(Wang et al., 2010)	———
Hybrid IDS	(A Collaborative Intrusion, n.d.; Ficco et al., 2016)	(Al Haddad et al., 2016; Chiba et al., 2016; Singh et al., 2016)	(Ahram et al., 2020; Arjunan & Modi, 2017; Balamurugan & Saravanan, 2019; Jung & Zarrabi, 2017; Mishra et al., 2017; Patil, 2018; Patil et al., 2019)

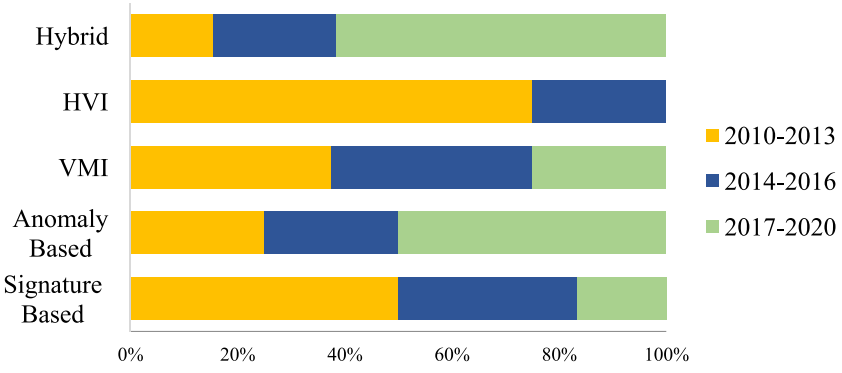

Fig. 7. Summary of the survey based on technique used.

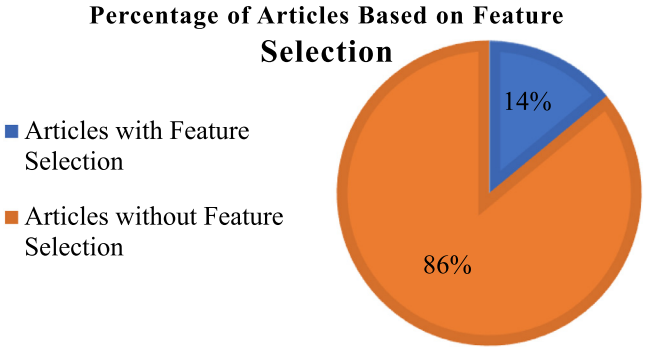

Fig. 8. Analysis of articles based on feature selection techniques.

Al-kasassbeh (2018), Sakr (2019), Pacheco et al. (2020), Intelligence et al. (2020) used feature selection methods during the period 2018–2020. Fig. 8 shows that only 14% of the total articles discussed used feature selection techniques. The results of these articles show that the technique presented using feature selection has fewer false alarms and an improved detection rate. Hence, in future research work, feature selection techniques should be considered to improve the system performance and accuracy. In addition, the efficiency of the system can be improved by using optimization techniques.

5.3. Observations

With the growth of big data, security becomes an integral part of cloud computing. Therefore, the development of intrusion detection systems has become a crucial research topic in cloud computing. IDS helps in the detection of malicious activities to secure the system, which motivates us to find issues in the present approaches so that one can work on them to resolve them. In this work, we have analysed 43 articles related to cloud security between

January 2010 and June 2020. We have selected articles from reputed journal and conferences such as Science Direct, Springer, Scopus, Research Gate, IEEE and some another international journal. It is found

Table 8
Comparison with related surveys.

Parameters	(Modi et al., 2013b)	(Khan, 2016)	(Alhenaki et al., 2019)	(Jebamalar et al., 1882)	(Azeez et al., 2020)	Present Work (2021)
Feature selection	Ï	Ï	Ï	Ï	Ï	P
Technique Description	P	P	Ï	P	P	P
Review of existing IDS techniques	P	P	Ï	Ï	Ï	P
VMI/HVI Technique	Ï	Ï	Ï	Ï	Ï	P
Research gaps and Future scope	P	P	Ï	Ï	Ï	P
Dataset used for Performance Evaluation	Ï	Ï	Ï	Ï	Ï	Ï

that a variety of security techniques are used to solve different security issues and that all techniques are efficient in one way or another. Each technique has merits and demerits. In this section we are presenting an observational summary of our work. This would be useful to understand the effectiveness and shortcoming of existing techniques used in earlier days of cloud security and how new techniques were developed to enhance the security.

- Signature detection is easy to use, but there is a need for database maintenance, increasing the storage requirement. It performs very well for known attacks that are stored in the database but cannot detect unknown attacks and zero-day attack. A tiny change in the attack program cannot be identified by this technique. To solve these issues, Anomaly detection was proposed.
- In anomaly detection, user behaviour is observed, and the system is trained according to that. Any deviation from the behaviour is considered as an attack. Moreover, if users add more services to their account or shift from one cloud to another, then this behaviour change cannot be identified by the system. It results in increased false alarms.
- In both the above technique, the security tool is placed on a monitored machine. If malware analysis is done in the tenant's VM, then advanced malware programs can compromise the analysis component thus making it easy to breach the monitored VM security. This requires the development of a technique to solve this problem. One solution is to isolate the IDS from the monitoring VM. Hence, deploying a security tool at Virtual Machine Manager (VMM) can serve the purpose. This technique of intrusion detection is known as virtual machine introspection (VMI), where virtual machines are monitored from outside the system.
- The success of the VMI technique depended on the assumption that the hypervisor is a secure place to run the VMI tool. However, a compromised hypervisor results in VM attacks managed by that hypervisor. So, to overcome this problem, the HVI technique was proposed, that deploys a security tool below hypervisor level. This technique examines the control and noncontrol flow data, memory, hypercalls, and data structure related to the hypervisor. But this technique requires hypervisor modification and is highly dependant on hardware design.
- From the survey, it's clear that individual techniques do not provide security to the cloud. Therefore, the researcher started using the hybrid technique. Compared to individual technique, hybrid IDS perform well and can detect various attacks. However, hybrid techniques can increase the system overhead and overall time complexity. Hence there is need to develop a technique that can provide cloud security from various attacks without increasing the system overhead. In future research, the feature reduction technique can reduce the system overhead and improve detection accuracy.

5.4. Comparison with related surveys

In Section 3, we have discussed many surveys published in the field of cloud security. We have selected five surveys from 2010 to 2020 for a comparative analysis of the present work. It is clear from the Table 8, that:

- Most of the surveys classified IDS technique into three broad categories like; signature-based IDS, anomaly-based IDS and hybrid techniques without consideration of specialized techniques like VM Introspection (VMI) and Hypervisor Introspection (HVI). without considering the HVI and VMI approaches of intrusion detection.
- No survey during 2010 to 2020 discussed the importance of feature selection in IDS.
- Moreover, open issues related to cloud security and future research directions are rarely discussed in previous surveys.

Thus, we analysed cloud security based on the four key elements of cloud security: Description of Attacks, Analysis of existing Intrusion Detection techniques, importance of Feature Selection in ID, dataset used for performance evaluation, and discussion of special IDS like Virtual Machine Introspection and Hypervisor Introspection.

6. Open issues and future research directions

In this section, we offer major cloud security issues that need to be explored. Many techniques have been developed to provide cloud security, but there exists scope for continuing and enhancing security techniques, as there are many cloud issues and challenges that need to be addressed to improve cloud security and privacy. For example, hierarchical approach of intrusion detection uses multiple IDS units to provide security to cloud environment. These units can be compromised and require an additional protection mechanism (Arjunan & Modi, 2017). Some techniques are developed for specific cloud, that cannot be used in a distributed environment like a cloud (Balamurugan & Saravanan, 2019; Mishra et al., 2017). Many researchers developed a cooperative network intrusion detection system that increases the overall communication overhead (Al Haddad et al., 2016; Chiba et al., 2016). However, many approaches performed well, but they need to be tested against recent attacks (Al Haddad et al., 2016; Arjunan & Modi, 2017; Balamurugan & Saravanan, 2019; Ficco et al., 2016; Zhang et al., 2020).

Another major challenge for cloud computing is to manage big data. Recently, a large volume of data is produced at rapid rate that need to be stored and pre-processed before further use. Using this data directly for intrusion detection can degrade the system's performance. Hence, a technique that can reduce the volume of data for intrusion detection is needed. One such technique is feature selection methods that can reduce the false alarm rate and improve the system accuracy. Our work showed that only a few articles considered feature selection techniques to reduce feature dimensionality (Intelligence et al., 2020; Pacheco et al., 2020; Patil et al., 2019; Prasad et al., 2019; Rawashdeh & Al-kasassbeh, 2018; Sakr, 2019; Zhang et al., 2020). Feature selection technique to handle big data in intrusion detection system, is quite promising.

Moreover, there is great concern about where to deploy the IDS unit, if it is deployed on VM (Alhenaki et al., 2019; Gupta & Kumar, 2015; Wolthusen, 2012), then the number of units will increase and it becomes computationally costly to handle a large number of IDS units. To reduce the number of units, it can be deployed on vSwitch or on a physical switch. But if an IDS unit is installed on vSwitch, it cannot handle heavy network traffic from the cloud environment and requires fast detection capability (Arjunan & Modi, 2017; Balamurugan & Saravanan, 2019; Mishra et al., 2017). Similarly, if it is deployed over a physical switch,

then it cannot monitor VM-to-VM traffic as a result insider attack cannot be detected. Developing an IDS technique that considers these factors is very desirable. To improve cloud security without increasing the cost, we will summarize important points related to the intrusion detection system for future research work.

High network traffic: With the growth of network devices, the cloud network faces a high volume of traffic, so the IDS should be able to handle this.

Fast detection: Speed and accuracy are major requirements of and security system that should be considered while developing an IDS. For this purpose, feature selection and optimization techniques can be used. These techniques improve the accuracy and speed of detection by reducing the false alarm rate.

Recent attack detection: The cloud subscribers and providers are regularly reporting the security issues and attacks. An IDS should be able to detect recent attacks.

Resistance to compromise: security of IDS in cloud is a critical issue. Hence, the security of the IDS must be considered.

Evasion Detection: Catching intrusions masked by evasion strategies is a difficult task for most IDSs. The efficiency of an IDS against evasion is determined by its ability to build an attack signature that can detect altered attacks. The effectiveness of IDS against various evasion tactics should be investigated further.

IDS deployment: The deployment of IDS is a major issue in cloud security. The deployment of IDS on each virtual machine can improve security, but also increase cost and overhead. Similarly, deployment of IDS on hypervisor, physical switch or at vSwitch reduces the cost and management overhead but makes collecting all context information of the monitored machine difficult. Hence, IDS placement should be done in such a way that it reduces the cost and improves the security.

Software aided solutions: Most of the IDS techniques used in the literature are hardware-based. A hardware-based technique requires modification in the system hardware and domain experts to deploy a security mechanism in the cloud. Furthermore, these techniques are less flexible (Patil et al., 2019). This issue can be handled by developing a software-based security technique

Datasets: As malware behaviours evolve, older datasets will no longer be effective over time, necessitating the development of newer and more complete datasets that cover a wider range of malicious activities.

7. Conclusion and future scope

In this work, we have discussed various security issues and malicious activities in different service models of cloud computing. These activities result in data breaches, account hijacking, malware injection, insider attacks, database manipulations, and flooding attacks (DoS/DDoS). Hence, Security in the cloud is a development of policies and procedures to protect cloud-based data, applications and infrastructure from unauthorized access by providing Confidentiality, Integrity, Availability and in time intrusion detection. For all these reasons, intrusion detection system (IDS) has been developed to identify suspicious activities and intrusions. In this work, we have analysed 43 articles of existing intrusion detection techniques and presented a classification model based on the IDS configuration, its placement, and attacks they detect. Articles between January 2010 and June 2020 are selected, and their merits and demerits are highlighted. We have discussed: the requirement of cloud security, issues and factors affecting cloud security, attacks in all service models (IaaS, PaaS, and SaaS) and datasets for performance evaluation of cloud IDS. To improve the overall performance of the system, we have also discussed the importance of feature selection and dimensionality reduction in the intrusion detection process. A comparative analysis of various existing IDS techniques is given with their advantages and limitations. Finally, some open issues have been discussed that need to be addressed in future work. Miscellaneous IDS have been developed in literature using different techniques. But with the increase in malicious activities, it is difficult to include all aspects of security simultaneously, so there is always an opportunity for improvements. For this reason, research should focus on improving intrusion detection by combining different techniques. Future research work should include feature selection and optimization techniques to improve system performance and detection accuracy. Additionally, to improve the cloud security without increasing the cost, we have also discussed various points for future research work.

Declaration of Competing Interest

None.

References

A Collaborative Intrusion Detection and prevention system in cloud computing, (2022).

A.K.M. A, Virtual machine introspection based spurious process detection in virtualized cloud computing environment, (2016).

Ahram, T., Karwowski, W., Vergnano, A., & Leali, F. (2020).Advances in intelligent systems and computing 1131 intelligent human systems integratio, 2020.

Al Haddad, Z., Hanoune, M., & Mamouni, A. (2016), A collaborative network intrusion detection system (C-NIDS) in cloud computing, 8 2016.

Aldribi, A., Traoré, I., Moa, B., & Nwamuo, O. (2020).Computers & security hypervisor-based cloud intrusion detection through online multivariate statistical change tracking, 88 10.1016/j.cose.2019.101646.

Aldwairi, M. (2017).Pattern matching of signature-based IDS using Myers algorithm under MapReduce framework, 10.1186/s13635-017-0062-7.

Alhenaki, L., Alwatban, A., Alamri, B., & Alarifi, N. (2019). A survey on the security of cloud computing. In *Proceedings of the 2nd international conference on computer applications and information security ICCAIS 2019* (pp. 1–7). 10.1109/CAIS.2019.8769497.

Arjunan, K., & Modi, C. N. (2017). An enhanced intrusion detection framework for securing network layer of cloud computing. In *Proceedings of the ISEA conference on Asia-pacific security, ISEASP 2017*. 10.1109/ISEASP.2017.7976988.

Azeez, N. A., Bada, T. M., Misra, S., Adewumi, A., Van der Vyver, C., & Ahuja, R. (2020). Intrusion detection and prevention systems: An updated review. *Advances in Intelligent Systems and Computing, 1042*, 685–696. 10.1007/978-981-32-9949-8_48.

Balamurugan, V., & Saravanan, R. (2019). Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation. *Cluster Computing, 22*, 13027–13039. 10.1007/s10586-017-1187-7.

Benninger, C., Neville, S. W., Yazir, Y. O., Matthews, C., & Coady, Y. (2012). Maitland, lighter-weight VM introspection to support cyber-security in the cloud. In *Proceedings of the IEEE 5th international conference on cloud computing CLOUD 2012* (pp. 471–478). 10.1109/CLOUD.2012.145.

Bharadwaja, S., Sun, W., Niamat, M., & Shen, F. (2011). Collabra, A xen hypervisor based collaborative intrusion detection system. In *Proceedings of the 8th international conference on information technology - new generations ITNG 2011* (pp. 695–700). 10.1109/ITNG.2011.123.

Borisaniya, B., & Patel, D. (2019). Towards virtual machine introspection based security framework for cloud. *Sādhanā, 44*, 1–15. 10.1007/s12046-018-1016-6.2013.

Charband, Y., & Navimipour, N.J. (2016).Online knowledge sharing mechanisms: A systematic review of the state-of-the-art literature and recommendations for future, 10.1007/s10796-016-9628-z.

Chiba, Z., Abghour, N., Moussaid, K., El Omri, A., & Rida, M. (2016). A cooperative and hybrid network intrusion detection framework in cloud computing based on snort and optimized back propagation neural network. *Procedia Computer Science, 83*, 1200–1206. 10.1016/j.procs.2016.04.249.

Citation, E.S. (2016).A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015, 0–21.

Denz, R., & Taylor, S. (2013). A survey on securing the virtual cloud. *Journal of Cloud Computing, 2*, 1–9. 10.1186/2192-113X-2-17.

Deshpande, P., Sharma, S.C., Peddoju, S.K., & Junaid, S. (2014).HIDS, A host based intrusion detection system for cloud computing environment, 10.1007/s13198-014-0277-7.

Ding, B., He, Y., Wu, Y., & Lin, Y. (2013). HyperVerify: A VM-assisted architecture for monitoring hypervisor non-control data. In *Proceedings of the 7th international conference on software security and reliability companion, SERE-C 2013. 1* (pp. 26–35). 10.1109/SERE-C.2013.20.

Ficco, M., Tasquier, L., & Aversa, R. (2016). Intrusion detection in federated clouds. *International Journal of Computational Science and Engineering, 13*, 219–232. 10.1504/IJCSE.2016.078929.

Gupta, S., & Kumar, P. (2015). An immediate system call sequence based approach for detecting malicious program executions in cloud environment. *Wireless Personal Communications., 81*, 405–425. 10.1007/s11277-014-2136-x.

Hofmeyr, S. A., Forrest, S., & Somayaji, A. (1998). Intrusion detection using sequences of system calls. *Journal of Computer Security, 6*, 151–180. 10.3233/JCS-980109.

Intelligence, A., Applications, N., Barolli, W.L., Moscato, F., Enokido, T., Takizawa, M., Villano, U. (2020).NOTICE : This is a pre-copyedited version of a contribution published in Web, 2L-ZED-IDS : A two-level anomaly detector for multiple attack classes,(n.d.).

Jebamalar, J. P. A., Paul, S., & Latha, D. P. P. (1882). *Mining classification algorithms : A comparative study*. Singapore: Springer. 10.1007/978-981-13-1882-5.

Jia, L., Zhu, M., & Tu, B. (2017).T-VMI , trusted virtual machine introspection in cloud environments, 10.1109/CCGRID.2017.48.

Jung, J., & Zarrabi, H. (2017).HIDCC : A hybrid intrusion detection approach in cloud computing, 10.1002/cpe.4171.

Khalil, I.M., Khreishah, A., & Azeem, M. (2014).Cloud computing security: A survey, 1–35. 10.3390/computers3010001.

Khammassi, C., & Krichen, S. (2017). A GA-LR wrapper approach for feature selection in network intrusion detection. *Computer Security, 70*, 255–277. 10.1016/j.cose.2017.06.005.

Khan, M. A. (2016). A survey of security issues for cloud computing. *Journal of Network and Computer Applications, 71*, 11–29. 10.1016/j.jnca.2016.05.010.

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurit, 2*. 10.1186/s42400-019-0038-7.

Kirat, D., Vigna, G., Kruegel, C., Vigna, G., & Kruegel, C. (2014).Sec14-paper-kirat.Pdf.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology, 51*, 7–15. 10.1016/j.infsof.2008.09.009.

Kumar, P., Nitin, V. S., Shah, K., Shukla, S. S. P., & Chauhan, D. S. (2011). A novel approach for security in cloud computing using hidden Markov model and clustering. In *Proceedings of the world congress on information technology WICT 2011* (pp. 810–815). 10.1109/WICT.2011.6141351.

Laurén, S. (2018).Virtual machine introspection based cloud monitoring platform,104–109.

Lengyel, T. K., Maresca, S., Payne, B. D., Webster, G. D., Vogl, S., & Kiayias, A. (2014). Scalability, fidelity and stealth in the DRAKVUF dynamic malware analysis system. In *Proceedings of the ACM international conference proceeding series. 2014-Decem* (pp. 386–395). 10.1145/2664243.2664252.

Lin, C. H., Tien, C. W., & Pao, H. K. (2012). Efficient and effective NIDS for cloud virtualization environment. In *Proceedings of the 4th IEEE international conference on cloud computing technology* (pp. 249–254). 10.1109/CloudCom.2012.6427583.

Lo, C. C., Huang, C. C., & Ku, J. (2010). A cooperative intrusion detection system framework for cloud computing networks. In *Proceedings of the international conference on parallel processing work* (pp. 280–284). 10.1109/ICPPW.2010.46.

Maiero, C., & Miculan, M. (2011). Unobservable intrusion detection based on call traces in paravirtualized systems. In *Proceedings of the international conference on security, privacy, and applied cryptography* (pp. 300–306). 10.5220/0003521003000306.

Mandal, J. K., Satapathy, S. C., Sanyal, M. K., Sarkar, P. P., & Mukhopadhyay, A. (2015). Information systems design and intelligent applications: Proceedings of second international conference India 2015. volume 1. *Advances in Intelligent Systems and Computing.*, 339. 10.1007/978-81-322-2250-7.

Marinos, L. (2013).ENISA threat landscape 2013 - Overview of current and emerging cyber-threats, 10.2788/14231.

Meng, Y., Li, W., & Kwok, L. F. (2014). Design of cloud-based parallel exclusive signature matching model in intrusion detection. In *Proceedings of the IEEE high performance computing and communications* (pp. 175–182). HPCC 2013 2013 IEEE Int. Conf. Embed. Ubiquitous Comput. EUC 2013. 10.1109/HPCC.and.EUC.2013.34.

Mimiso, M. (2012, September). Virtual machine escape exploit targets xen. ⟨http://threatpost.com/virtual-machine-escape-exploit-targets-xen-090612/76979⟩.

Mishra, P., Pilli, E. S., Varadharajan, V., & Tupakula, U. (2017). Out-VM monitoring for malicious network packet detection in cloud. In *Proceedings of the ISEA Asia security and privacy conference, ISEASP*. 10.1109/ISEASP.2017.7976995.

Mishra, P., Verma, I., Gupta, S., & Rana, V. S. (2019). vProVal , Introspection based process validation for detecting malware in KVM-based cloud environment. In *Proceedings of the 4th international conference on fog and mobile edge computing* (pp. 271–277). 10.1109/FMEC.2019.8795365.

Modi, C., Patel, D., Borisaniya, B., Patel, A., & Rajarajan, M. (2013a). A survey on security issues and solutions at different layers of cloud computing. *Journal of Supercomputing, 63*, 561–592. 10.1007/s11227-012-0831-5.

Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., & Rajarajan, M. (2013b). A survey of intrusion detection techniques in cloud. *Journal of Network and Computer Applications, 36*, 42–57. 10.1016/j.jnca.2012.05.003.

More, A., & Tapaswi, S. (2014).Virtual machine introspection: Towards bridging the semantic gap.

Moustafa, N., & Slay, J. (2015).UNSW-NB15 , A comprehensive data set for network intrusion detection systems, 10.1109/MilCIS.2015.7348942.

Pacheco, J., Benitez, V. H., Filix-Herran, L. C., & Satam, P. (2020). Artificial neural networks based intrusion detection system for internet of things fog nodes. *IEEE Access: Practical Innovations Open Solutions* 1–1. 10.1109/access.2020.2988055.

Pandeeswari, N., & Kumar, G. (2015).Anomaly detection system in cloud environment using fuzzy clustering based ANN, doi:10.1007/s11036-015-0644-x.

Patil, R. (2018).Protocol specific multi-threaded network intrusion detection system (PM-NIDS) for DoS /DDoS attack detection in cloud.

Patil, R., Dudeja, H., & Modi, C. (2019). Designing an efficient security framework for detecting intrusions in virtual network of cloud computing. *Computer Security, 85*, 402–422. 10.1016/j.cose.2019.05.016.

Pfoh, J., Schneider, C., & Eckert, C. (2022).A formal model for virtual machine introspection.

Prabadevi, B., Jeyanthi, N., & Abraham, A. (2020). An analysis of security solutions for ARP poisoning attacks and its effects on medical computing. *Journal of Systems Assurance Engineering and Management*, 11. 10.1007/s13198-019-00919-1.

Prasad, M., Tripathi, S., & Dahal, K. (2019), urn a, Applied Soft Computing Journal, 105980. 10.1016/j.asoc.2019.105980.

Rawashdeh, A., & Al-kasassbeh, M. (2018).An anomaly-based approach for DDoS attack detection in cloud environment an anomaly-based approach for DDoS attack detection in cloud environment Adnan Rawashdeh * Mouhammd Alkasassbeh and Muna Al-hawawreh, 10.1504/IJCAT.2018.093533.

Roesch, M. (2015).Snort – lightweight intrusion detection for networks, 229–238. http://www.usenix.org.

Rong, C., Nguyen, S. T., & Jaatun, M. G. (2013). Beyond lightning: A survey on security challenges in cloud computing. *Computers and Electrical Engineering, 39*, 47–54. 10.1016/j.compeleceng.2012.04.015.

SyedNavaz, A., Sangeetha, V., & Prabhadevi, C. (2013). Entropy based anomaly detection system to prevent DDoS attacks in cloud. *International Journal of Computers and Applications, 62*, 42–47. 10.5120/10160-5084.

Sakr, M.M. (2019).Network intrusion detection system based PSO- SVM for cloud computing, 22–29. 10.5815/ijcnis.2019.03.04.

Santoso, B.I., Idrus, M.R.S., & Gunawan, I.P. (2016), Designing network intrusion and detection system using signature-based method for protecting openstack private cloud.

Sari, A. (2015).A review of anomaly detection systems in cloud networks and survey of cloud security measures in cloud storage applications, 142–154.

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the ICISSP 2018 - international conference on information systems security and privacy 2018-janua* (pp. 108–116). 10.5220/0006639801080116.

Shi, J., Yang, Y., & Tang, C. (2016). Hardware assisted hypervisor introspection. *SpringerPlus*. 10.1186/s40064-016-2257-7.

Singh, D., Patel, D., Borisaniya, B., & Modi, C. (2016). Collaborative IDS framework for cloud. *International Journal of Network, 18*, 699–709.

Srinivasan, M. K., Sarukesi, K., Rodrigues, P., Manoj, M. S., & Revathy, P. (2012). State-of-the-art cloud computing security taxonomies - A classification of security challenges in the present cloud computing environment. *ACM International Conference Proceeding Series*, 470–476. 10.1145/2345396.2345474.

Subhy, M., & Basheer, D. (2018).A comparison study for intrusion database (KDD99, NSL-KDD) based on self organization map (SOM) artificial neural network.

Symantec, (2015). Symantec intelligence report. https://www.symantec.com/.../intelligence-report-06-2015.en-us.pdf).

Thampi, S.M., Sherly, E., Dasgupta, S., Lloret, J., Abawajy, J.H., & Khorov, E. (2019).Lecture notes in networks and systems 125 applied soft computing and communication networks.

Wang, J., Stavrou, A., & Ghosh, A. (2010).HyperCheck, A hardware-assisted integrity monitor, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 6307 LNCS 158–177. 10.1007/978-3-642-15512-3_9.

Wang, Z., Wu, C., Grace, M., & Jiang, X. (2012). Isolating commodity hosted hypervisors with hyperlock. In *Proceedings of the EuroSys 2012 conference* (pp. 127–140). 10.1145/2168836.2168850.

Wolthusen, S. D. (2012). Detecting anomalies in IaaS environments through virtual machine host system call analysis. In *Proceedings of the international journal of internet technology and secured trans London* (pp. 211–218).

Yuxin, D., Xuebing, Y., Di, Z., Li, D., & Zhanchao, A. (2011). Feature representation and selection in malicious code detection methods based on static system calls. *Computers & Security, 30*, 514–524. 10.1016/j.cose.2011.05.007.

Zhang, Y., Juels, A., Reiter, M. K., & Ristenpart, T. (2012). Cross-VM side channels and their use to extract private keys. In *Proceedings of the ACM conference on computer and communications security* (pp. 305–316). 10.1145/2382196.2382230.

Zhang, Z., Wen, J. I. E., Zhang, J., Cai, X., & Xie, L. (2020). A many objective-based feature selection model for anomaly detection in cloud environment. *IEEE Access: Practical Innovations Open Solutions, 8*, 60218–60231. 10.1109/ACCESS.2020.2981373.

Zhou, M., Zhang, R., Xie, W., Qian, W., & Zhou, A. (2010). Security and privacy in cloud computing, A survey. In *Proceedings of the 6th international conference on semantic computing knowledge grid, SKG* (pp. 105–112). 10.1109/SKG.2010.19.