IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 9, 2020, accepted June 1, 2020, date of publication June 3, 2020, date of current version June 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2999715

A Review of Intrusion Detection and Blockchain Applications in the Cloud: Approaches, Challenges and Solutions

OSAMA ALKADI, (Member, IEEE), NOUR MOUSTAFA, (Senior Member, IEEE), AND BENJAMIN TURNBULL

School of Engineering and Information Technology, University of New South Wales at ADFA, Canberra, ACT 2612, Australia

Corresponding author: Nour Moustafa (nour.moustafa@unsw.edu.au)

This work was supported by the Australian Postgraduate Award (APA).

ABSTRACT This paper reviews the background and related studies in the areas of cloud systems, intrusion detection and blockchain applications against cyber attacks. This work aims to discuss collaborative anomaly detection systems for discovering insider and outsider attacks from cloud centres, including the technologies of virtualisation and containerisation, along with trusting intrusion detection and cloud systems using blockchain. Moreover, the ability to detect such malicious attacks is critical for conducting necessary mitigation, at an early stage, to minimise the impact of disruption and restore cloud operations and their live migration processes. This paper presents an overview of cloud architecture and categorises potential state-of-the-art security events based on their occurrence at different cloud deployment models. Network Intrusion Detection Systems (NIDS) in the cloud, involving types of classification and common detection approaches, are also described. Collaborative NIDSs for cloud-based blockchain applications are also explained to demonstrate how blockchain can address challenges related to data privacy and trust management. A summary of the research challenges and future research directions in these fields is also explained.

INDEX TERMS Intrusion detection systems, collaborative anomaly detection, cloud systems, blockchain applications, approaches, challenges, solutions.

I. INTRODUCTION

Cloud systems face sophisticated attack scenarios that has increased with the emergence of blockchain. For instance, in June 2018, several blockchain cryptocurrencies, including Bitcoin Gold, Zencash and MonaCoin, all fell victim to a 51% attack, leading to loss of over 18 million worth of tokens [1]. The attackers exploited each cryptocurrency network and temporarily gained more than half of the total global mining hash rate for each currency, effectively centralizing the decentralized systems [2]. Blockchain-based applications have emerged in multiple domains to offer trust and data privacy services. Blockchain offers new opportunities by allowing participants to exchange transactions and share information while maintaining a degree of trust, integrity and enhanced transparency. Blockchain technology has numerous applications across different domains that go beyond the financial services and digital currency [3], including the energy sector [4], Internet of Things (IoT) [5], supply chain and manufacturing [6], privacy preservation [7], big data [8], and anomaly detection [9].

Intrusion Detection Systems (IDSs) [5], [8], [10] and blockchain solutions [2], [4] have been applied to cloud systems to identify cyberattacks and protect private data, respectively [4]. IDSs in the cloud are effectively classified based on deployment locations, and are categorized as host-based or network-based [8], [11]. A Host-based IDS (HIDS) runs on a host system or Virtual Machine (VM) to monitor and inspect audit data of operating systems, including memory and process audits [5], [12]. If the HIDS detects a malicious activity from an individual host or VM, the source IP is defined as access to the whole network to prevent user-to-root attacks from VM hopping and gaining access to another VM. A Network-based IDS (NIDS) is placed at the infrastructure layer of enterprise, or increasingly cloud networks to monitor network traffic of all connected systems within a subnet [13].

The associate editor coordinating the review of this manuscript and approving it for publication was Haider Abbas.

One of the primary concerns in the cloud is the ability to maintain data protection and trust management between multi-cloud service providers [14]. Cloud systems are public, distributed and decentralised, and this potentially leads to challenges of trust as different components are controlled by different parties. Cloud providers are usually reluctant to share data or report intrusion events due to concerns about data confidentiality and privacy [8], [15]. It is quite difficult to measure the level of reputation among untrusted participants. A collaborative IDS (CIDS) would be a protection layer to detect insider and outsider attacks, which denotes the development of distributed intrusion detection engines across network nodes of cloud systems [11]. It should be scalable and cost-effective to inspect various cloud nodes for discovering new cyber attacks. Another major challenge is insider attacks such as collusion and betrayal attacks, where malicious nodes collaborate to give false information and degrade the efficiency of alarm aggregation [16]. In addition to the protection layer of CIDS, blockchain can assist in trusting cloud nodes and their distributed intrusion detection engines.

Several surveys in the literature have separately reviewed intrusion detection and blockchain technologies. On the one hand, for intrusion detection, the authors reviewed IDSs and their attributes, including detection techniques, IDS deployment strategies, security threats, and validation strategies in IoT systems [17]. Moustafa *et al.* [18] discussed cyber kill chain models and cyberattacks that would breach network and cloud systems. Besides, this work reviewed multiple Decision Engine (DE) approaches, involving new ensemble learning and deep learning algorithms. Sharam and Kaul [19] explained the concept of IDSs and provided more details of IDSs that have been utilised in Vehicular Ad-hoc Network (VANETs) and VANET cloud systems. Furthermore, others have proposed a deep blockchain framework designed to offer security-based distributed intrusion detection and privacy-based blockchain with smart contracts in the cloud. The intrusion detection method was employed using a Bidirectional Long Short-Term Memory deep learning algorithm to deal with sequential network data and was evaluated using real-world datasets for classifying attack events that exploit cloud networks [20].

On the other hand, for blockchain, Xie *et al.* [21] reviewed the background of blockchain technology, and how this technology would be employed in smart cities, from multiple aspects, including that of smart citizens, smart healthcare, smart grid, smart transportation, and supply chain management. In [22], the authors described existing blockchain techniques in the domain of IoT applications. They also explained the challenges of implementing blockchain technologies and possible solutions and future research directions in this domain. More recently, Xie *et al.* [23] presented an overview of utilising blockchain for cloud exchange. They then briefly surveyed blockchain technology and explained the challenges of using blockchain for cloud exchange in the perspectives of security and privacy.

In this paper, we present a comprehensive review of IDS and blockchain and how they could be used together to offer security and privacy perspectives in cloud systems. The main contributions of this paper are to give more details of cloud systems, and IDSs including their detection methods, deployments in the cloud, including live migration process, virtualisation and containerisation, as well as types of attacks. Also, we describe blockchain technology and attack families that would exploit them. Finally, we discuss the main challenges of using IDS and blockchain and their possible solutions, and future research directions in these domains.

The rest of the paper is structured as follows. Section II outlines the current state of cloud computing. Section III discusses virtualisation, containerisation and live migration. Following that, the current taxonomies of cyber security attacks and threats in the cloud are described in Section V. Section VI outlines current and emerging approaches to cyber security, including IDS and blockchain in the cloud. Moreover, blockchain and smart contracts and their security threats within the cloud systems are explained. Section VIII discusses the current state of security in cloud systems, and this is followed by the challenges and future research directions in Section IX. Finally, section IX-A provides the paper summary of our review.

II. CLOUD COMPUTING SYSTEMS

This section begins with a brief introduction to cloud computing with an emphasis on related security architecture concepts used in this work. Additionally, state-of-the-art cloud security threats and attacks that can be used to perform malicious events in cloud computing systems are summarised.

A. CLOUD COMPUTING CONCEPTS AND ARCHITECTURE

Cloud computing is a general term for delivering IT-based services from multiple geographic locations over the Internet [24]. These virtual shared resources include software applications, operating systems and network infrastructure that are accessible, anytime and anywhere in the world [25]. Cloud services have been widely utilised for both business and personal use through social media and email communications. In addition, cloud storage solutions, such as Amazon Web Services (AWS), Microsoft Azure Blob and Google Cloud Storage, are broadly used to access documents across different devices.

The cloud allows organisations to rapidly provision computing resources based on actual demand, reducing costs and simplifying the consumption of IT infrastructure with minimal management effort or third-party vendor interaction [26], [27]. As shown in Figure 1, a common cloud computing architecture includes a set of loosely coupled services, for example, software, platform, and infrastructure–as–a–service, whereby each service refers to the availability of a particular resource for ensuring continuous operation without interruption or data loss [27]. Recent advances in reliable high-speed networking enable the rapid growth of cloud-based services, which have led to the prevalence of

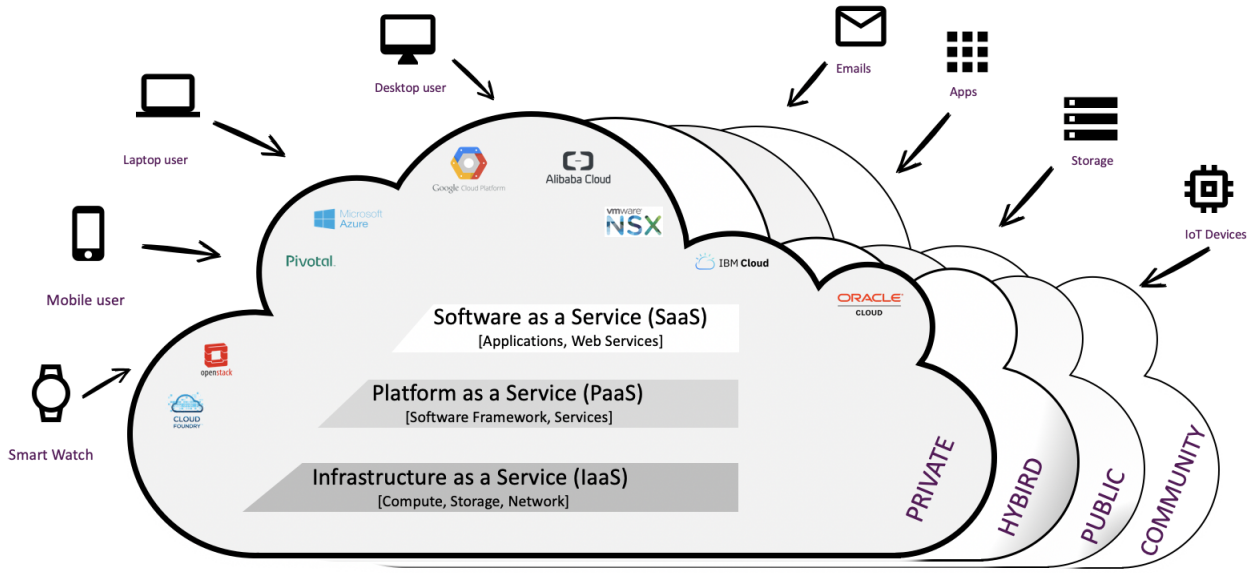

FIGURE 1. Cloud computing architecture.

virtualisation, containerisation, Service-Oriented Architecture (SOA), and utility computing. Each of these is discussed as follows.

- *Virtualisation* refers to the process of creating a virtual representation of computing resources by emulating physical systems to run operating systems and end-user applications. The technology is a fundamental element of cloud implementations, which delivers essential cloud features of resource pooling, location independence, and rapid elasticity. The advantages of this technology can also reduce compatibility issues between different hardware platforms, operating systems or network resources [28]. At a hardware level, several physical resources, including CPUs, memory, hard drives and network devices, are located across distributed datacentres, which are responsible for processing and storage requirements. Above this layer, there are the software, hypervisors and management layers that permit the effective running across servers and gateways [28]. A hypervisor is a computer program in the host Operating System (OS) that runs multiple guest OS within it. The management layer can monitor peaks in traffic and auto-scale to meet the demands of changing workloads by controlling provisioning state of VMs. It also has the capability to manage security policies and access control rules across the cloud environment.
- *Containerisation* is strongly related to virtualisation, and refers to the concept of creating one or more virtualised runtime environments on a single OS. It allows implementing software and packages its code and dependencies in a standard unit, the so-called container, which offers significant efficiencies in computation, space, complexity and deployment times [29]. One of the most common containers is a docker image, which is a flexible, lightweight and executable package of software. It involves the entire dependencies, libraries, and configuration settings to implement an application. Every container runs a specific software application, which denotes that it is an abstraction of the application layer, running isolated processes on an OS [30]. However, there are significant issues in protecting containers against cyber attacks, where a single kernel exploit would endanger other processes in the kernel of the operating system [31]. More detail of containerisation and its security challenges is presented in Section IV.
- *SOA* is a paradigm for the design and implementation of software solutions, where services are provided to other application components through a communication protocol over a network. Cloud computing, combined with SOA, has influenced business process components to be assembled and orchestrated more efficiently to deliver distinctive business services and capabilities for higher performance [32].
- *Utility computing* is a delivery model in which a service provider makes computing resources and infrastructure management to consumers on-demand as metered service. An example of this is grid computing, seeking to maximise the efficient use of resources and minimise their associated costs [33].Cloud computing has primarily emerged from grid computing and offers on-demand resource provisioning. Grid computing involves the use of multiple computing resources to solve a single task or a large-scale problem that is too complex for a single machine to handle. The key differences between cloud and grid computing are the use of visualisation and deployment model, which maximises available computing power. Virtualisation can separate the logical components from the physical artifacts of computer systems,

solving some of the issues faced by grid computing. Although grid computing accomplishes high utilisation by allocating multiple resources to provide a running environment to solve a single task, cloud replies on a single virtualised resource to compute several tasks simultaneously [28], [34].

The general consensus among researchers, regarding similarities between the two paradigms, imply that cloud computing builds on grid computing. Hence, it is considered the foundation for cloud computing. Furthermore, both computing types are scalable through load balancing of the application running separately on different operating systems and connected with web services. Compute instances and network bandwidth can be provisioned or deallocated as needed. Their storage capacity rises and drops based on the running instances, the number of users, as well as the amount of data transferred at a time [28].

B. CHARACTERISTICS AND DEPLOYMENT MODELS OF CLOUD COMPUTING

Cloud computing is considered one of the most popular implementations of distributed computing. In 2006, Amazon Web Services (AWS) became the first cloud provider to promote an alternative to on-premises infrastructure by offering computing and network storage services [35]. Several other vendors have since entered this market, with AWS, such as Google Cloud Platform (GCP) and Microsoft Azure are also the two dominant players in the public cloud space. The initial offering of compute, storage and networking provided by these vendors has also significantly expanded, and modern cloud providers offer significantly more complex and diverse services. These include IoT platforms, containers, serverless platforms, business intelligence, ephemeral computing functions, Software Defined Networking (SDN) paradigms and much more [26]. All public cloud providers are expected to share six common characteristics of cloud computing: deployment flexibility, broad network access, location independence, infrastructure scalability, sustainability and reliability [28], [35].

Deployment flexibility allows a cloud user to allocate computing resources, such as storage volumes and software, through a cloud host provider automatically without requiring human interaction. This is typically based on an agreed subscription or pay–as–you–go scenario. Broad network access refers to available resources by the cloud provider and accessibility through multiple device types, such as laptops and workstations, but also includes mobile platforms. In essence, for as long as a cloud consumer can access their required services, this characteristic will be met. Sometimes enabling broad network access may rise some operational concerns such as complex monitoring, traceability, and auditability. Location independence is when a cloud provider offers computing resources, such as compute, networks, and storage, that are pooled to serve different consumers, whereby each is isolated using a multi-tenant model, which are all accessed through the internet. Such resources reply on virtualisation technologies; therefore, generally, the consumer does not know exact physical locations of provided services since VM gets dynamically assigned and frequently migrated between different datacentres, all within the purview of the cloud provider.

Cloud providers have the responsibility to provide efficient allocated resources in order to satisfy the Service Level Agreement (SLA) and achieve data privacy of their cloud consumers requirements. Infrastructure scalability is the ability to provide consumers to automatically provision and scale services rapidly on demand. To the cloud user, such a feature appears to be infinite and sometimes appropriate in any quantity at any time. Sustainability of distrusted technology can be achieved due to the nature of the SOA approach utilised in cloud computing, the amount of cloud resource usage by the consumer can be automatically and dynamically provisioned, monitored and controlled, which creates a transparency between the Cloud provider and consumer. Reliability in cloud computing can provide high availability by deploying applications across multiple data centres in different zones and regions. This reliability makes cloud computing a suitable platform for disaster recovery and business continuity.

Cloud computing architecture is still an evolving paradigm. This presents various unique security challenges. It also raises the issue of how data stored on third-party vendors, can be securely protected [3], [6]. Cloud security is seen as a shared responsibility between consumers and cloud providers, depending on delivery models. Cloud services are classified into three service models: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [26], [34], [35]. Figure 2 illustrates the cloud service layered architecture and share security model through which software, platform, and infrastructure are provided to consumers based on the pay on-demand model, as described below.

- *Software as a Service (SaaS)* - a cloud service where consumers can only run and access software applications through web portals. The provider manages all underplaying cloud infrastructure. SaaS applications are usually provided on a pay as you go or subscription-based service.
- *Platform as a Service (PaaS)* - software developers and DevOps teams are able to deploy their applications onto the cloud infrastructure, without being concerned about installing any platforms or setting up network's services. However, developers should be aware that changes to system configuration or even updates in the PaaS components may introduce application-specific security vulnerability.
- *Infrastructure as a Service (IaaS)* - it provides virtualised computing resources over the Internet such as storage, networks, or resource provisioning on demand. Usually, the client has full control and access to the virtualised Cloud components, including OS, storage and deployed applications.

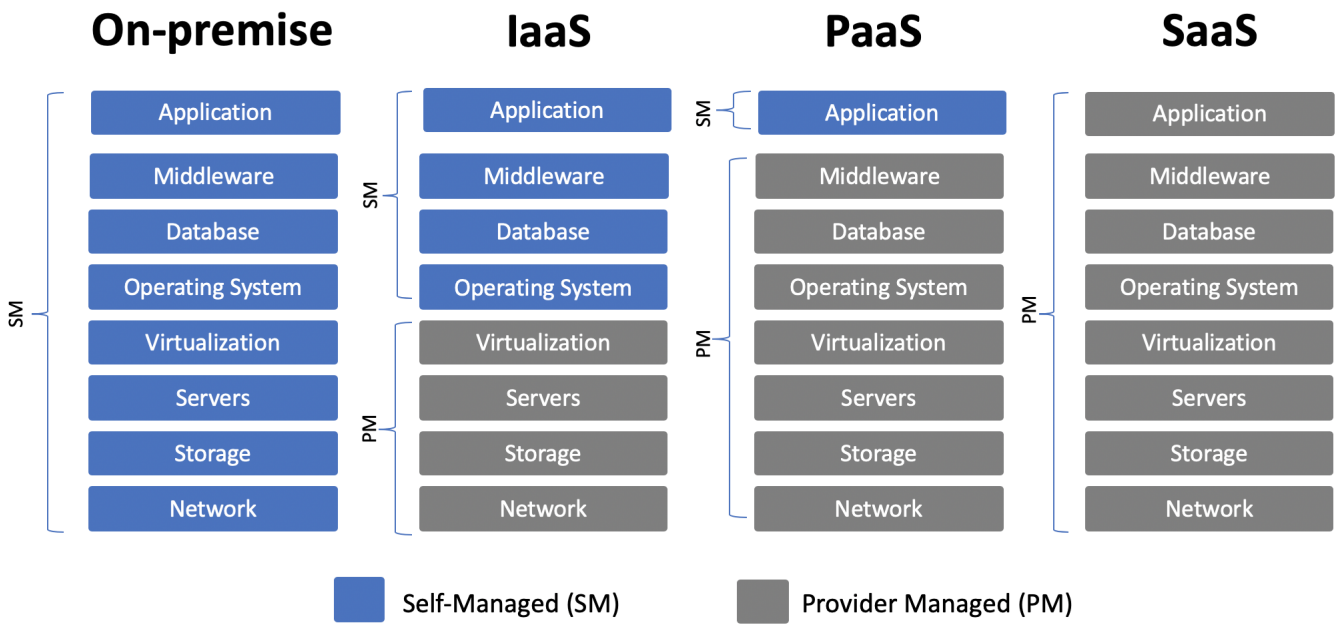

FIGURE 2. Cloud service comparison and shared security model.

The adaptive and dynamic provisioning architecture of cloud services raise serious concerns for cloud consumers regarding the manner in which the SLA of their infrastructure can be managed. In particular, when hosted applications and their data is being migrated between different CSP as most providers only guarantee availability but fail to meet performance and data privacy requirements [36]. Therefore, failing to achieve such goals will risk the sustainable growth of cloud computing in the future and may result in critical applications bring run on-premise.

The concept of cloud deployment describes how services are made available to consumers and generally categorised on the basis of security, size and data sovereignty. There are four types of cloud deployment models; private, public, community, and hybrid infrastructure, as illustrated in Figure 1 [34]. Each of these is discussed separately. Private cloud infrastructure is dedicated to virtualised resources and a secure environment for a high-profile organisation, which may exist on or off premises and could be managed by multiple consumers. This model provides greater control and privacy. Public cloud infrastructure that is created using a pool of shared resources, which is publicly available and accessible by multiple tenants. This model may only exist in the cloud provider's own infrastructure.

According to a recent study by International Data Corporation (IDC), in 2019 the annual yearly spending on public cloud services is expected to grow by 22.5% to reach $370 Billion in 2022 [37]. Community cloud infrastructure that is designed for a limited group of consumers sharing a common interest. This is usually managed and governed by all participating groups which may exist on- or off-premises. Hybrid cloud infrastructure which is a mixture of one or more private and public clouds to perform distinct services within the same organisation. The hybrid model gives consumers greater flexibility by allowing workloads to be distributed between private and public clouds as computing needs and cost change. These distinctions are likely to change over time. For example, the use of accredited classified public cloud changes the paradigm. In reality, these systems might not be multi-tenanted, but it would be difficult for a user to make this distinction.

III. VIRTUALISATION AND LIVE MIGRATION

As discussed, virtualisation is a fundamental component in delivering as-a-service capabilities for cloud-based solutions. By providing a combination of resource sharing, security, server isolation, and live migration, virtualisation allows a single physical server to run several applications in isolated environments by creating VMs using VM Monitor (VMM or hypervisor) to manage access control [38] as shown in Figure 3. To enforce isolation mechanisms between VMs, virtualisation platforms must achieve key requirements such as security, fault, performance and software isolation [39]. Security isolation prevents one malicious guest VMs from compromising other guest VMs applications and data, by using standard security features such as class loaders in Java, or code access security in .NET [40]. Fault isolation is designed to prevent workload failures or application bugs from propagating to other VMs and gaining unauthorized access to system resources. There must be effective performance isolation between co-hosted VMs through scheduling and resource allocation. Finally, software isolation allows each VM to run its own dedicated operating system and applications.

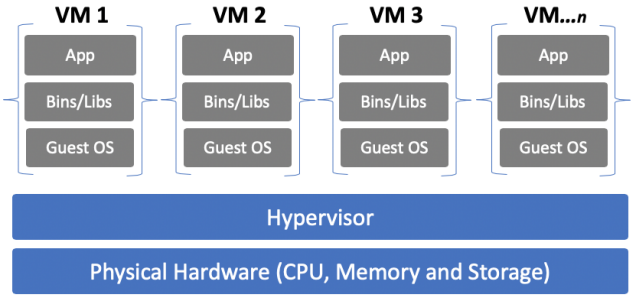

FIGURE 3. Hardware virtualisation.

The live migration in cloud systems refers to the process of moving a virtual machine state and system properties from one host source to another host target while maintaining continues service [41]. All VM configurations, storage, operating system, and application state remain unchanged. The live migration goal is to maintain quality of service by minimising the total migration downtime, through pre-negotiated set of SLAs. More recently, hosted hypervisors such as Red Hat KVM, Citrix XenServer, Microsoft Hyper-V, and VMware vSphere have begun offering VM migration in real-time [42]. This service keeps VMs running in the event of hardware failure, infrastructure maintenance, BIOS upgrade, and emergency security patching, allowing one host to fail over and still maintain uptime on hosted VMs.

VM memory migration is the term used for live migration; it involves moving the memory of the VM from one host to another, preserving its state and giving the appearance of remaining 'live'. There are two main techniques to achieve VM Memory migration; pre-migration and post-migration [43]. The pre-migration memory is the preferred method for most hypervisors and is made up of two iterative phases; push and stop-and-copy. During the push phase, all guest memory pages are transferred to the destination host while the VM is still executing on the source. If any changes occur to memory pages during the copy process, they need to be retransmitted to maintain consistency across the two VM instances. These changes are also referred to as dirty memory pages. The pre-migration process will enter the stop phase when the rate of dirty pages drops below a specified maximum threshold, i.e., the correlation between the total migration time and expected downtime. In the stop-and-copy phase, the VM is first suspended on the original host, any remaining inconsistent dirty pages are copied over, and the new host VM is started.

The brief moment when both VMs are not running is called 'black-out' or 'down-time' and ranging from milliseconds to a few seconds, depending on the guest VM memory size and running applications. If the migration process is unsuccessful, the process will be repeated. There are several techniques to reduce the total migration time, such as work presented in [41], [44]–[46]. In the post-copy migration method, the source host is suspended, the VMs minimal processor state is copied to target host and only on completion the service is resumed on a target host. If the migration process is unsuccessful in case of failure or exceeding SLA, the process may be repeated again. This migration method has a higher period of downtime; however, it is considered to be less complex. Achieving successful complex migration is depended on the performance of the VMs. Workloads on virtual and physical machines can be very memory intensive and might require additional infrastructure. This unexpected high demand for cloud resources might introduce a form of hotspot on datacentres that exceed provisioned capacity [47], [48].

Hotspots occur quickly, and host migration needs to occur faster than can be manually detected and initiated. Usually, this is dealt with by either increasing the resources allocated to a virtual server or by migrating the virtual server itself to another datacentre. However, the manual operation is not robust enough to overcome sudden changes in workloads, and errors may arise since multiple virtual servers are involved in each migration to redistribute the system load. Therefore, automated procedures, orchestration and optimised schedules are required for the detection of hotspots and effective responsiveness within the agreed SLA [49]. While the previously mentioned characteristics of cloud platforms create numerous benefits for cloud consumers, they also introduce new forms of cloud-specific attacks that are an attractive target for potential intruders. These attacks are currently both technically diïČcult and economically expensive [50], but have the potential to cause high-impact results. These types of attacks increase the security vulnerabilities and allow cyber intruders to preform various malicious activates, as summarised in the following section.

IV. CONTAINERISATION FOR LIGHTWEIGHT VIRTUALISATION

As outlined, there has been a dramatic shift in the field of virtualisation platforms with the introduction and steady release of container-based virtualisation technologies such as Docker [51], LXC [52] and CoreOS Rocket [29]. Containers provide an alternative lightweight and microservice-based architecture to traditional server virtualisation deployments. Next generation cloud native applications are being developed, packaged and deployed using DevOps-based continuous delivery model [53]. However, Container orchestration over clusters of nodes in large dynamic cloud datacentres still face major challenges in terms of flexibility and efficiency [30].

Containers are often compared to VMs, since both of the virtualisation solutions allow multiple isolated services to be deployed in confined environments. A key distinction between VMs and containers, the latter is an abstraction of the underlying host OS, allowing the application with its dependent libraries to run on multiple isolated environments as shown in Figure 4. Furthermore, this framework enables interoperability of software application packaging with on-premise clusters or between multi-cloud vendors [53]. The main limitation of this approach in live migration scenarios is considered less flexible, since all guests

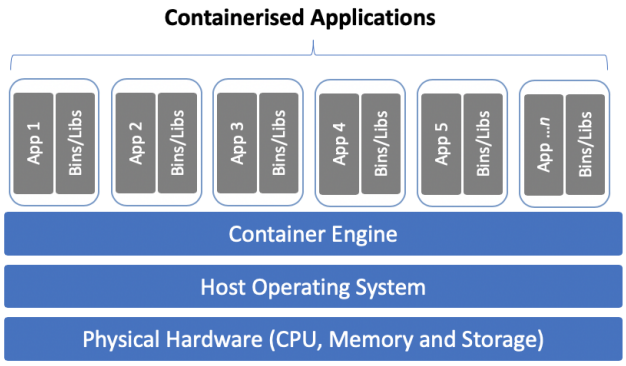

FIGURE 4. Deployment of containers.

require to run the exact kernel version to host OS. However, in hypervisor-based approaches, which abstracts physical hardware, each VM image includes a full copy of guest OS. As a result, containers can deploy more applications with minimal VMs requirements, providing efficient utilisation of resource allocation and potentially reduction in management overhead [29]. There is, however, an overhead in data that must be transferred. The overall differences and similarities between the two technologies can be seen in Table 1.

Container-based applications inherit same classes of threats as traditional VM deployments as well as the potential for unique security challenges. Attacks include: DoS, ARP poisoning, MAC floods and unwanted cross–talk between containerised systems [31], [54]–[57]. Container solutions reply on key Linux kernel security features such as namespaces and cgroups to handle isolation processes, access control and shared resource allocations [58]. However, a number of security vulnerabilities and malware were found in both official and community container templates as they lack any automated security patching processes [59], [60]. This is partially mitigated by the expected ephemeral nature of such systems, but in practice this is not always the case.

V. CYBER SECURITY ATTACKS AND TAXONOMIES IN CLOUD COMPUTING

The cloud offers significant benefits in scalability, agility, transparency, and availability. However, security is often considered an impediment to these. Although a consideration, it is much more difficult to quantify, and is context-dependent within itself [61]. The centralisation of data on cloud offers security controls and governance capabilities that would protect consumer privacy and meet regulatory compliance. Furthermore, the homogeneity and automation nature of the cloud enables cloud providers to direct their security capabilities on securing cloud services [28]. However, these advantages present new cloud-specific vulnerabilities inherited from the underlying technologies. These technologies include virtualisation, APIs, containers and datacentres. In many cases, it is not a vulnerability it of itself, but a misconfiguration or technological design decision being misinterpreted or incorrectly implemented. Any of these could threaten organisations adopting the cloud architecture. Additionally, new vulnerabilities in the specification or implementation of technologies that underpin cloud may also have an impact on cloud vendors or implementations.

The cloud service model comprises three interdependent layers of architectural construct; IaaS, PaaS and SaaS. Each layer is susceptible to specific vulnerabilities introduced by either misconfiguration or malicious intent of consumer or service provider. Cloud systems can be breached by many cyber adversaries, exposing the three computer security principles of Confidentiality, Integrity, and Availability (CIA) often referred to as the CIA triad [62] of its resources and services. Under this circumstance, data and virtualised infrastructure could be compromised by existing and new forms of attacks. Therefore, the security challenges of cloud systems become a significant concern when cloud resources and data privacy are abused by an insider or outsider intruder [63].

A. CLOUD SECURITY THREATS

Cyber threats to information security and infrastructure residing in the cloud vary according to the type of delivery model used by cloud consumers. The Cloud Security Alliance (CSA) top frequent cloud threats include data breaches, Advanced Persistent Threats (APT) and Denial of Service (DoS) [64]. This is shown in Figure 5. The CSA has also produced security best practices and recommendations for critical areas of focus in cloud architecture, governance, virtualisation and containers and various other security operations. Similarly, the Open Web Application Security Project (OWASP) describes top ten cloud security risks as well as most critical web application security flaws, particularly when deploying cloud-based solutions or SaaS models [65]. The recent data breach investigation report by Verizon communications, analysed almost 42,000 cyber security incidents and over 2,000 confirmed data breaches from 86 countries targeted cloud solutions [66]. Overall, the outsider threats remain predominant and accounting for 69% of breaches, while insider perpetrated incidents were the remaining 31%. Moreover, the 2019 MacAfee annual threat report, revealed ransomware attacks grew by 118%, followed by PowerShell and crypto mining attacks attempting to compromise cloud accounts [67].

Cloud and virtualisation technologies are also vulnerable to traditional network attacks, given that they are services on existing internet architectures and platforms. This includes attacks against existing network protocols, such as Address Resolution Protocol (ARP), Border Gateway Protocol (BGP), and Domain Name Service (DNS). These attacks target protocols that are globally actively used, but were designed and implemented before security was a concern, and hence have few, if any, mechanisms for ensuring authentication, security, validation, or encryption. ARP spoofing, BGP hijacking, and DNS poisoning have been used in the real world [68]. Such attacks also include Distributed Denial of Service (DDoS), and backdoor attacks. Distributed Denial of Service is an attack designed to cause resource exhaustion on the

TABLE 1. Features of virtual machines (VMs) and containers.

Features	Virtual Machines	Containers
Security and isolation	Provide a strong degree of isolation from the host OS and other VMs. This is more suitable for mission critical systems where security boundaries are crucial	Provide lightweight spatial isolation, as they share the same Kernel with other containers.
Operation System	Each VM runs a full OS version which can be different from host's OS. Requires more system resources.	Only runs the user space portion of the OS and requires less system resources.
Deployment	Deployment is substantially lengthy as different instances are responsible for execution.	Easy deployment and interoperability across different environments
Memory Efficiency	Less efficient as it requires the entire OS to be loaded including a full boot process.	No disk space is required, hence less memory.
Fault tolerance	Failover clusters are used in case of hardware failure without downtime	Container failed cluster node can be automated re-deployed to another cluster using orchestration platforms
Usage	Demanding mission critical applications and network infrastructure.	Continuous integration and deployment of Web applications and small databases

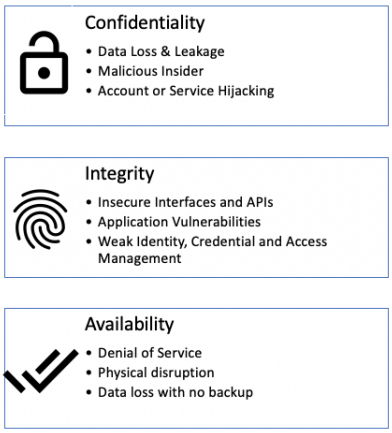

FIGURE 5. Recent cyber security threats in cloud systems.

target node, but extends upon DoS by utilising numerous nodes [69]. The combined bandwidth, memory or ability to open connections by thousands, or tens of thousands of distributed nodes can overwhelm many systems. In many of these cases the attacking nodes are themselves unaware and have been previously been compromised [64].

The internet makes this form of attack more likely, given the global reach of connectivity. For example, in 2019 AWS infrastructure suffered a sustained series of DDoS attacks on its route 53 infrastructure, raising questions about AWS Shield DDoS protection services being implemented [70]. Insider attacks are another form of adversary that, whilst not unique to cloud-based systems, are of concern and can potentially cause significant impact. Insiders in cloud systems come in two forms; the malicious employee or contractor, and the adjacent tenant in multi-tenanted systems. The identification and mitigation of the former of these is considered a current area of cyber security research [71], and the impact was shown with the leaks from Edward Snowden in his role as a US contractor [72], [73]. The second class of insider, adjacent tenants on multi-tenanted cloud systems, is a cloud-specific issue. Adjacent tenants have higher-capacity to flood network traffic, breach systems and can attempt to subvert a system through accessing its underlying hypervisor. Whilst modern firewall implementations are able to provide an array of protection against outsider attacks, they are not designed to prevent insider attacks [68].

To ensure a secure cloud infrastructure, multiple layers of defensive technologies are required. Specifically, there is a need for an adaptive Intrusion Detection System (IDS) that can cleverly detect anomalies and attacks in a network, and moreover be able to differentiate between the various scenarios encountered when performing a network behaviour analysis. There are several common types of insider and outsider attacks that attempt to compromise the CIA of cloud resources and services. These are detailed as follows [68]. Insider intruder, or insider threat, refers to a potential individual who is a current or former employee, including external contractors of an organisation gaining unauthorised access to its cloud resources or services to cause significant harm or disruptions. They could disclose information, modify important information, or commit frauds either maliciously or unintentionally in a way that could negatively impact business operations. A typical example of this incident can occur when a database is misconfigured by a system administrator, inadvertently exposing sensitive client information to the public.

A VMM attack is an attack against the hypervisor. When a hypervisor is compromised using zero-day attacks, a cyber attacker is able to gain control of the installed VMs and everything that is associated with physical hosts can be manipulated [74]. These attacks occur before cloud vendors had an opportunity to provide a security patch or made aware of the vulnerability. There are two types of hypervisor attacks; through the host OS or guest OS. Host OS attack is designed to take advantage of underlying hypervisor infrastructure and gain control of the host operating system, by exploiting vulnerabilities and security holes in an unpatched hypervisor. Once attackers have administrator or root access, the hypervisor is compromised along with all other active running VMs. Similarly, compromised guest OS (also known as VM Escape), allows the attackers to gain unauthorised access to another VM on the same physical host by exploiting some weak hypervisor vulnerability, resulting in damaging

multiple virtual server-based websites. A Flooding attack is a form of denial of service attack that attempts to flood victims by sending a succession and enormous packet requests from malicious 'zombie' computer hosts. These packets could be TCP/IP, UDP, IGMP, ICMP or a combination of them. In a cloud computing architecture, requests for VMs are accessed by any user on the Internet, which might cause a DoS/DDoS attack via zombies.

When an attacker exposes a single server, this might lead to a total loss of service availability and render it unusable. If a cloud hardware resource capacity is entirely exhausted by processing the flood requests, all other VM instances on the same physical machine would not be able to complete their tasks. The aftermath of a flood attack might cause a significant usage spike in bill charges, as cloud providers might not be able to discriminate between normal and abnormal usage. The two popular types of flooding attacks in the cloud are DoS and DDoS attacks. The DoS attacks use only one computer to launch attacks to overload target's processing power while the DDoS attacks use multiple systems to target a single cloud application, web server and database vulnerabilities. Cloud service abuses are malicious activities can hijack cloud computing resources by taking advantage of free trials offers and fraudulent payment registrations. For example, attackers might use legitimate cloud services to lunch malware, DDoS and brute-force attacks, distribute spam and phishing emails, crypto mining and hosting malicious content. Thereby resulting in a reduction in available resources, loss in productivity and loss of data for legitimate customers hosted CSP. Any reputation damage incurred will be against the cloud provider, rather than the malicious user or users.

The APT is a form of cyber attack [64] in which an intruder gains unauthorised access to computing infrastructure and remains undetected for an extended period of time. APTs are also considered a class of threat, as they are considered well-resourced and motivated. The goal of the APT attack is to establish a footprint then stealthily monitor network activity and steal intellectual property rather than cause damage to cloud systems. Intruders explicitly go through a great deal of effort to carry out APT attacks, hence they target highly protected and valued information within large organisations such as government agencies, defence contractors and financial institutions. APTs have travelled across cloud systems and blend in with legitimate network traffic which makes it difficult to identify and detect. Most could providers apply proactive security solutions to combat APT attacks from compromising their infrastructure. However, it is critical to educate cloud users to share the same level of diligence in protecting their cloud accounts as they would do in on-premises systems. Some complex APTs require dedicated teams of cyber security experts to maintain the compromised cloud systems and software, which lead to increase spending and indirect economic damage.

Common threats of APTs include advanced exploits of zero-day vulnerabilities, direct hacking attacks, spear phishing, penetrating compromised third-party networks, and USB drives preloaded with malware. Security awareness programs and regular training are the most effective methods in keeping users informed and ready to combat such attacks. Port scanning attack is a method to determine a list of open, closed and filtered ports in a network (no reply or an ICMP error). Although the technique itself is not inherently hostile, it is often used as part of the reconnaissance phase to gather information about vulnerabilities of targeted systems. Attackers probe servers or hosts for all active ports to reveal the presence of security devices such as firewalls and eventually to find loopholes gaining unauthorised access to a system. Discovery and mapping of available network services, including router, gateway filtering, firewall rules, IP and MAC address, could be easily exposed by this attack. There are several port scanning mechanisms, for example, TCP scanning, SYN scanning, Xmas and FIN scanning, ACK scanning, UDP scanning, Windows scanning, used to launch this attack. Cloud computing infrastructure is also vulnerable to this type of attack, targeting its shared networks and VMs.

Insecure interfaces and APIs occur when Cloud providers offer web related services that can be accessed through a set of software programming interfaces or APIs. The most common API styles are based on protocols such as HTTP/REST, JSON/XML, SOAP and WSDL. Therefore, the security of the hosted services depends entirely upon the security of these APIs. A weak set of credentials or inappropriate input data validation may expose organisations to various security threats. Additionally, the context in which individuals, developers or users set up and maintain their APIs potentially expose additional avenues of attack. This form of attack is often context-dependent. Backdoor attacks are a form of passive attack that employs covert methods to bypass a standard authentication or encryption mechanism to gain access to privileged systems or corrupt data without detection. Typical backdoor capabilities allow an attacker to control a victim's resources and use it as a zombie to perform DDoS attacks. Backdoor installation is accomplished by taking advantage of vulnerable applications, and it is usually hard to detect since files tend to be highly concealed. It will potentially lead to several malicious actives, including; data theft, APT assaults, and eventually taking control of the whole system. For instance, a lessor type of backdoor attack is keylogging technique to track every keystroke on the victim machine, including capturing screen activates, including system credentials. Similarly, in cloud environments, an attacker can get access and control cloud user's resources via the backdoor channel and make VMs as a zombie to launch DoS/DDoS attacks.

User to Root (U2R) attacks occur when an intruder tries to gains access to legitimate user account by sniffing, dictionary and social engineering attacks. Consequently, leading to exploit vulnerabilities for obtaining the root level access of the victim's account. For instance, Buffer overflows are used to make root shells from a service running as root. It occurs when a software bug or malicious script attempts to overfill data in a static buffer than it can handle, causing

TABLE 2. Insider and outsider attacks on cloud computing categorised by impact.

Attack Type	Properties	Service Impact	Examples
Insider intruders	A legitimate cloud user gaining unauthorised access to carry out malicious activities and cause significant harm.	IaaS, PaaS, SaaS	misconfiguration, Spy, Ransomware
Attacks on hypervisor	when the virtual layer of hypervisor is compromised using zero-day attacks, attackers can control the installed VMs and physical hosts	IaaS	VM Escape, VM, Hopping, Cross-VM Side Channel
Flooding attacks	Sending massive succession SYN packets for the purpose of disrupting cloud services.	IaaS	Neptune, Teardrop, Smurf
Services Abuses	It can be hijacked by malicious activities,,for example, using cloud/fog computing resources to violet an encryption key, to launch an attack.	IaaS, PaaS	DoS/DDoS, Cryptojacking, BGP Hijacking
Advanced persistent threats (APTs)	Penetrate systems to launch a foothold attack, then stealthily infiltrate data and intellectual property.	IaaS, PaaS, SaaS	Sykipot, GhostNet, APT28/29/34
Port scanning attacks	Scan servers and hosts to determine active ports resulting in identify potential vulnerabilities.	IaaS, PaaS	Portsweeping, Fin Scan, SYN Scan
Backdoor attack	Employs covert methods to bypass a standard authentication or encryption mechanism to gain access to privileged systems or corrupt data without detection.	IaaS, PaaS	SQL Slammer worms, Tuareg Viruses, Banker Trojans
User to Root (U2R) attacks	Gaining access to a legitimate user's account by exploiting vulnerabilities to obtain the root level access of a compromised system.	IaaS, PaaS	Load Module, Rootkit,Buffer Overflow

data to overflow into adjacent memory. Despite being well understood, there are no standard security methods to prevent buffer overflows, since applications fail to manage memory allocations and input validations. In cloud computing systems, attackers get access to valid user's accounts which enables them to gain root-level access to VMs or physical hosts. In this paper, we consider the attack taxonomy listed above to categorise insider and outsider attacks on cloud systems by their properties and relevant impact on each cloud service model as summarised in Table 2.

This section has summarised the current taxonomies of cyber security attacks and threats in Cloud Computing. The next section will discuss current and emerging approaches to combat cyber security threats related to cloud-based systems.

VI. CLOUD COMPUTING SECURITY SOLUTIONS

As outlined above, the highly configurable and scalable nature of cloud computing architectures potentially increases the threat of security vulnerabilities and loopholes that can be used to compromise cloud services. This section outlines current and emerging approaches to cyber security as related to cloud-based systems.

A. THE ROLE OF BIG DATA IN CLOUD SECURITY

cloud systems are scalable and operated at scale for end users. This scale imposes unique security requirements, vulnerabilities and opportunities, particularly in relation to big data [27]. The overwhelming amount of data being stored in cloud today, cannot be effectively processed using conventional methods. Big data is very different compared to standard databases where it is measured by key attributes such as volume, variety, velocity, value and veracity, and visibility. Analysis of extraction of useful information can aid organisation and CSP to improve their accurate identification of cyber threats or malicious activities. [75], [76]. Hence these 6Vs characteristics can facilitate making effective decisions for automated and scalable security solutions designed to protect cloud infrastructure, as described below:

- Volume is the sheer amount of generated data;
- Velocity is the flow rate of generated data;
- Veracity is the trustworthiness and availability of generated data;
- Variety is diversity and types of generated data;
- Value is the usefulness and benefits of generated data; and
- Visibility is the level of clarity and state of the generated state.

There are multiple technologies and processes designed to process and extract knowledge from big data, and it is an area that cloud providers are uniquely suited to operate. Given the scale of cloud deployments, the security data accessible is of interest if it can be processed effectively.

B. INTRUSION DETECTION IN CLOUD SYSTEMS

As more computer systems are being migrated to cloud infrastructure, their vulnerabilities are shared by the entire platform. This section will discuss concepts and recent related work on IDS, collaborative IDS, and blockchain in cloud systems and how this technology can be applied to solve challenges around data trust management. There are several existing security techniques, in different stages of technological maturity, designed to handle cyber security challenges as scale. These include application whitelisting, multi-factor authentication, restricting privileges, system patching, network monitoring through firewalls, IDSs and blockchain. However, in the case of complex cloud systems, there is no single mechanism to deal with all types of security threats. Therefore, to ensure a secure cloud infrastructure, multiple mechanisms should be combined to offer a comprehensive layer of defence.

Concept of IDS: The objective of an IDS is to provide a defensive layer against malicious activities that attempt to compromise cloud computing systems. Majority of IDS typically perform the following main functions [77]:

- Data recording and retrieval - IDS audit logs are usually collected and stored either in a centralised manner or sent to further analysis to security log management solutions such as Security Information and Event Management SIEM) or Data Loss Prevention (DLP). Archiving a complete collection of log records could enhance the investigative capabilities and help in meeting security compliance standards.
- Alter notifications - IDS alerts are essential for discovering potential threats and notify security operators for prompt remediation solutions. Alter notifications types include; emails, remote logging (e.g., Syslog), and Simple Network Management Protocol traps (SNMP). The massive volume of generated alters are usually challenging to manage. Hence, an important indication of how accurate intrusion detections by an IDS is to keep FN and FP alters to a minimum.
- Report generation - each IDS should have the capabilities to create actionable reports on monitored events that a company has deemed worthy of the internet.

An IDS can be either a hardware machine or a software application that monitors and analyses the network system for potential threat and policy violations. Current cloud cyber threats are becoming highly sophisticated and detecting such threats is both costly and time-consuming. Therefore, active IDS are crucial in preventing and securing IT cloud operations. As shown in Figure 6, the primary IDSs classifications are usually based on the location of deployment and detection methodology [78]. For example, a HIDS runs on each virtual machine to monitor and inspect inbound and outbound network packets. If it detects malicious traffic, then it may block the source IP access to the network to prevent user-to-root attacks from VM hopping and gaining access to another VM while a network-based IDS (NIDS) are placed at the infrastructure layer within IaaS to monitor traffic to and from all connected systems within the same physical host. It can identify direct and indirect flooding, backdoor, port-scanning attacks, and suspicious malware activities [13].

Moreover, IDS detection methods can be divided into four categories of triggering mechanism: signature, anomaly, stateful and hybrid-based merging multiple classes [16], [63], [79]. A Misuse-based Detection System (MDS) monitors network traffic and rely on a signature database to match observed behaviours or patterns of a known type of attacks and malicious activities. Although this type of detection has a lower occurrence of false-positive rates and higher detection rates to known attacks, there is a significant drawback in that it does not have the ability to detect zero-day or even previously unknown attacks. Therefore, cloud administrators must keep the signature database updated. This is considered an effective mechanism for known threats.

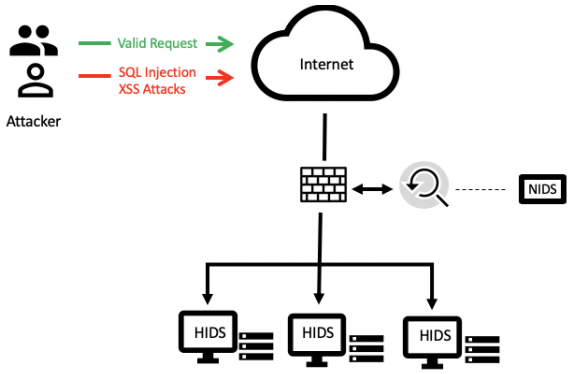

FIGURE 6. Deployment of both HIDS and NIDS in a cloud environment.

By contrast an Anomaly-based Detection System (ADS) uses a normal profile created by security cloud administrators to identify any deviation from it as an anomaly and alter the system. Unlike MDS, it has the ability to identify existing zero-day attacks; however, it still suffers from significant drawbacks, particularly in cloud computing environments. Firstly, it has a higher occurrence of false-positive rates as alters are generated anytime there is a miss-match from the normal profile. Secondly, determining normal traffic can be externally tricky and time consuming for the security administrators. Furthermore, a Stateful Protocol Analysis-based Detection System (SPADS) inspects variations of protocol state such as DNS, FTP and HTTP using a predefined universal profile in both network and application layers. It is then able to analyse and identify an unexpected or repeated sequence of commands. For instance, if a malicious user sends an unauthenticated FTP session, in this state the user can perform limited commands such as sending username and passwords.

SPADS can inspect the paring of request and response packets sent against state code to dreamtime any abnormalities. Although SPADS by itself is a very powerful technique, it is usually resource demanding generating excessive resource overhead. Additionally, it relies on cloud vendor-specific or standards bodies definitions, unlike AIDS which uses network-specific profiles. A Hybrid-based Detection System (HDS) integrates different detection methods into one system to overcome single limitation and improve detection of zero-day or even previous unknown attacks. For example, ADS and MDS detection methods can be combined to improve the overall performance of IDs [49], [50]. The main idea is that MDS would be detect existing attacks, whereas ADS detects unknown attacks. The hierarchy of these different forms of IDS are outlined in Figure 7. It shows that, the primary IDSs classifications are usually based on the location of deployment and detection methodology. Detection methods can be divided into four categories of triggering mechanism: signature, anomaly, stateful and hybrid-based merging multiple classes. Where is, the cloud deployed environment location can be host-based, network-based, hypervisor-based or collaborative-based combining multiple environment.

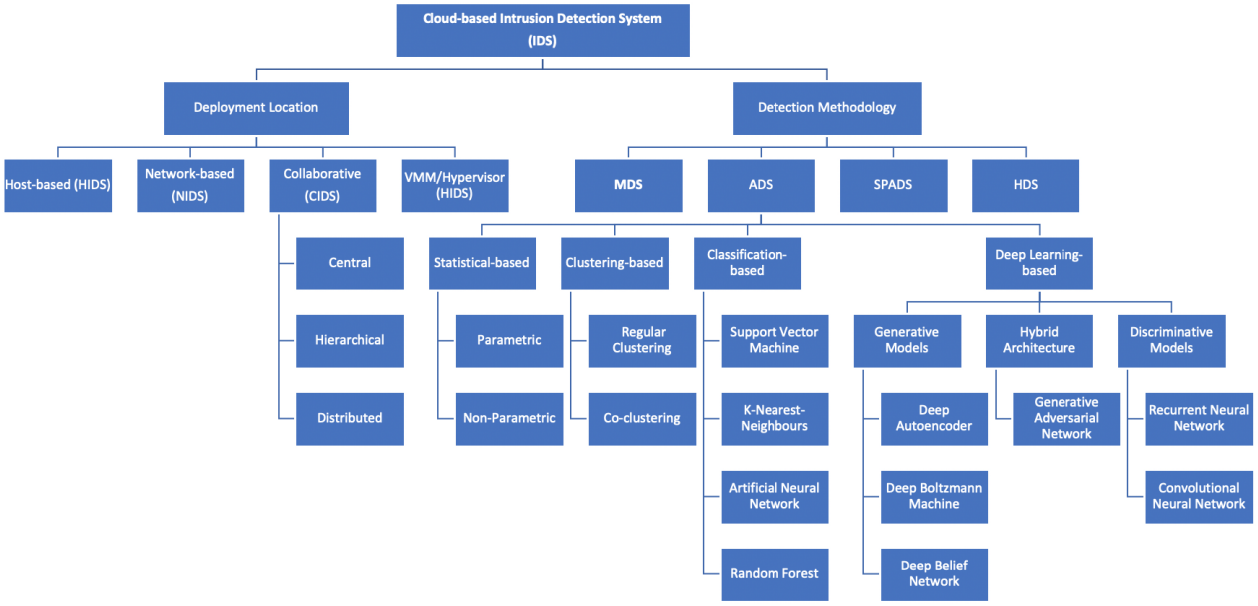

FIGURE 7. Intrusion Detection System classification methods.

Similar to traditional IDS on-premises, cloud deployed specific IDS can also discover malicious events either in real or offline detection. In real-time detection, attacks are detected while the host, VMs, or network is being monitored for abnormal activities and can immediately flag any deviations as an attack. Contrariwise, an offline detection handles audit trail with delay. Audit logs are collected and stored either in a centralised manner from a single source, or from distributed multiple locations. A VM monitor solution inserts as a software layer to control the physical resources, and it allows running many operating systems. It can improve the efficiency of detecting and preventing attacks in IDS as they have complete control of the system resources and complete visibility of the internal state of the monitored machines [80]. The advantages and disadvantages of IDSs in cloud systems are demonstrated in Table 3.

C. COLLABORATIVE IDSs IN CLOUD SYSTEMS

CIDSs consist of multiple IDSs deployed on large distributed networks or individual hosts that communicate with each other to detect coordinated cyber attacks. The primary purpose of CIDS is to enhance the overall detection accuracy of a single IDS node by correlating attack evidence over various sub-networks [63]. Thus, enforcing cooperation between different nodes would improve the capabilities to monitor sophisticated intrusions such as denial-of-service (DoS), distributed DoS (DDoS) and malicious insiders [85]. Furthermore, an isolated IDS would be easily bypassed by zero-day exploits or polymorphic code. Traditional CIDS can be dived into three communication overlay categories; centralised, hierarchical and hierarchical [84]. A centralised CID system comprises multiple physical servers or VMs that monitor prospective hosts or capture any traffic passing through. These systems share and exchange their locally collected data to a central coordinator unit that handles and analyses them. The main drawback of this implementation is the lake of scalability particularly in large scale networks, followed by redundancy and server arability. For example, Snapp *et al.* proposed one of the earlier centralised based CID systems called distributed intrusion detection system (DIDS), which combined distributed monitoring with a centralised data analysis [86].

A hierarchical CID system is divided into several smaller hierarchical structure groups based on features such as deployment model, physical locations and software platforms. Through this structure, hierarchical CID can mitigate the scalability by aggregating collected data to a higher level for further processing and analysis. However, this approach suffers from detection accuracy issues until data converge to high levels. For example, Moon *et al.* [78] proposed hierarchical CIDS called event monitoring enabling responses to anomalous live disturbances (EMERALD) designed to monitor malicious events across multiple domain layers in large enterprise networks. Other similar systems include SURFcert and CRIM [84]. A distributed CID system is independent, autonomous techniques that employ Peer-to-Peer (P2P) network architecture and therefore do not reply to the central coordinator unit for data analysis and correlation works. However, this approach still has its drawbacks and still suffers from detection accuracy as data might not be available from other participant hosts during the detection and analysis phase. Examples of P2P-based CIDS includes the distributed overlay for monitoring internet outbreaks system (DOMINO) developed by Yegneswaran *et al.* to counter insider attacks and fake alerts among heterogeneous

TABLE 3. Advantages and disadvantages of different classification of IDS in cloud systems.

Environments	Advantages/Strengths	Disadvantages /Limitations	Shared Responsibility
Network-based IDS (NIDS)	Portability and no degradation of performance as it only needs to read portion of each pack from its network segment [81].	Unable to analyse encrypted traffic data and can only capture part of the packet headers content [82]. Limited visibility within VMs as the usually deployed at the edge of a network [78].	Cloud Provider
Host-based IDS (HIDS)	No extra infrastructure needed. Monitors malicious activities within an organisational internal VMs against such as attempts to rewrite registry files or system settings [78].	Requires individual deployment on each VM and can only detect attacks on installed hosts.	On VMs and Hypervisor. Cloud providers and consumers
VMM Hypervisor-based IDS (VHIDS)	Visibility of information as it allows security administrators to monitor and analyse communications between multiple VMs, between hypervisor and VM, hypervisor and guest OS within the same sharedvirtual hardware [83]. Suitable for public cloud providers as it provides greater security control over hypervisor.	Proprietary software drivers might not work on existing server hardware causing potential compatibility issues and high running cost. New technology and requires specialist skills needed to maintain.	Deployment in a hypervisor. Cloud provider
Collaborative-based IDS (CIDS)	Integrates the characteristics of both HIDS and NIDS to overcome single limitations and improve detection performance. Ability to detect attacks patterns across the entire enterprise networks or across distributed cloud locations.	High computational cost and network overheads [84].	Deployment on VMs, hypervisors and external networks. Cloud and consumers.

autonomous systems [87]. Similarly, Pontarelli *et al.* [80] also proposed the P2P architecture of PIER, an internet-scale query processor that supports large distributed and continuous database-style query engines [88]. Pontarelli *et al.* [80] proposed a cloud-based CIDS framework for enhancing big data security, however, it did not provide a centralised correction handler, which made it susceptible to scalability issues.

Recently, deep learning-based techniques have been employed in various systems to design and improve the performance of IDSs [89]–[93]. These techniques allow systems to automatically learn feature representations needed for detection or classification from large unlabelled raw data [94]. Due to the increased demand for computational resources, recurrent and convolutional neural network algorithms are gaining increased attention and are recently applied in a supervised or unsupervised learning model for detecting anomalous events [95]–[97]. This is mainly attributed to their ability to find patterns from sequence data of cloud networks [98]. It is anticipated that deep learning-based approaches can help improve the overall performance and efficiency of cloud-based IDSs [99]–[101].

Modern CIDSs are built and deployed on could computing infrastructure because of their heterogeneous model and virtualised technology. Different cloud vendors may exchange data on malicious software activities and events logs among each other. However, if such SIEM products are not trusted and appropriately integrated, the practical usage of shared data becomes limited. The unique characteristics of could computing present several challenges when designing a cloud-based CID system. These desired characteristics include; efficient detection of insiders and outsiders' attacks while keeping FN and FP alters to a minimum. The ability to scale dynamically across different datacentre networks in the entire cloud. Furthermore, the framework should provide maximum security resistance to zero-day vulnerabilities to ensure data confidentiality, authentication, and integrity across all participant ICDS systems [27]. Figure 8 shows a typical architecture of a cloud-based collaborative intrusion detection network on how both HIDS and NIDS collaborate to implement instruction detection analysis at VM and network levels. Different IDS within the same cloud domain collaborate to share data or report intrusion events based on implicit trust. However, compromised or malicious nodes can provide false information and degrade the efficiency of alarm aggregation such as in case of collusion and betrayal attacks [14]. One emerging issue in IDS is how to preserve data privacy, and prevent attacks from altering data, logging, or transmissions during use the event of live data migration between multi cloud providers.

D. BLOCKCHAIN AND SMART CONTRACTS

The fundamental concept of a blockchain is that the development of a cryptographically protected method, which gains a publicly certifiable and immutable sequence of records, so-called, blocks. The blocks are ordered by timestamps, which are shared and synchronised through peer-to-peer networks [102]. The blockchain technology is utilised as a public, distributed ledger of transaction rows, that could be used to secure data transactions of cloud systems. Each user in blockchain networks can observe data blocks to verify or reject them using the consensus method. When a data block is accepted and verified, the block is inserted into the chains based on its timestamp [103]. The original theory behind the blockchain technology is that the principle of developing chaining records timestamped by using cryptographically secure hash functions. This theory was developed in the early

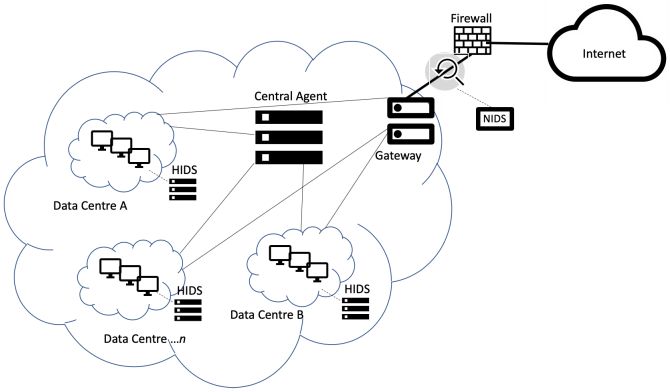

FIGURE 8. Cloud-based collaborative instruction detection system CIDS.

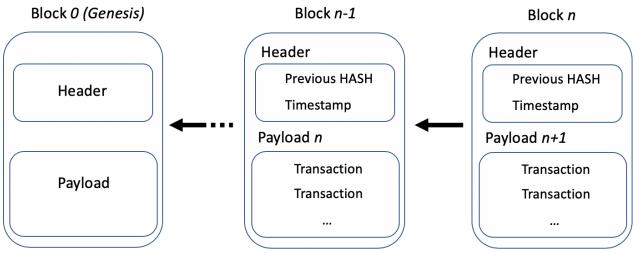

FIGURE 9. Elements of a blockchain.

1990s by Haber and Stornetta [69], and recently it is widely used in the Bitcoin cryptography [104]. Blockchains are considered secure cryptographic hash functions (H) that can map an arbitrary size input (i.e. message) to a fixed size n-bit output (i.e. a message digest), such that $\{0, 1\}* \rightarrow \{0, 1\}^n$.The contents of blockchain usually include the payload (i.e., a data block) and block metadata that involves a timestamp and a hash value of the last block in the chain, as depicted in Figure 9. It is observed that the timestamp typically offers a discrete-time value that gradually increases while extending the chain [69].

The development of hash functions in blockchain should consider some security requirements [102]. To start with, a first-layered preimage resistance refers to the difficulty of retrieving hash values given the hash function. In more detail, a predefined hash value (h) should demand $O(2^n)$ complexity to estimate an x, where $H(x) = h$. Secondly, a second-layered preimage resistance should be also considered, for example, an input (x) and its hash value ($h = H(x)$), it should demand $O(2^n)$ complexity to estimate an $x^0 = x$, where $H(x^0) = h$. Finally, a collision resistance of hashing should need $O(2^{n/2})$ complexity to estimate any two hash values ($x = x^0$), where $H(x) = H(x^0)$. This one-way cryptographic function should be resistant to first and second pre-imaging, and collision attacks. In the blockchain technology, particular importance should be given to second pre-imaging attacks since introducing a mid-fix would permit for altering the blocks while preserving the chain connected. Therefore, the size of the n-bit hash function should have a typical value of at least 512 bits in order to avoid attacks having a 2^n complexity [105].

1) TYPES OF BLOCKCHAIN

There are three common categories of blockchain-like implementations; public, private and hybrid (also known as a consortium). Each of these types is separately discussed. A public blockchain is an open and public blockchain where anybody can participate in the peer-to-peer network and is designed to be fully decentralised. This permissionless network model is based on an incentive process to encourage more individual participants to join the network. Examples of the public blockchain include Bitcoin and Ethereum, which are the largest blockchain-like implementations. However, this model requires a large amount of computational power to maintain the shared distributed ledger. Additionally, all participants must calculate the crypto mathematical problems known as Proof of Work (PoW) and Proof of Stake (PoS) for a consensus algorithm. Lastly, it lacks privacy and anonymity (pseudo-anonymous), since all transactions are publicly traceable.

A private blockchain is a closed and permission only blockchain model, where every participating node is privately invited and vetted by the network owner. This managed network will restrict access control for participates in two ways; existing participants decide future participants, or license issued by authority or consortium, such as Hyperledger and Quorum. In this scenario, providers collaborate and create their own network and self-maintain. This can be applied in organizations involved in credit card ratings, insurance companies, or secondary market of used vehicles. The motivation of a private model is not usually monetary and is set up within a private cloud network that allows only a few selected participants to verify and add transactions. However, transactions are still visible and traceable to all participants.

The hybrid blockchain model is a blend of both the public and private blockchain network models. Every node can operate privately in its own chain and only commit to a public network when verification is necessary. This makes it possible

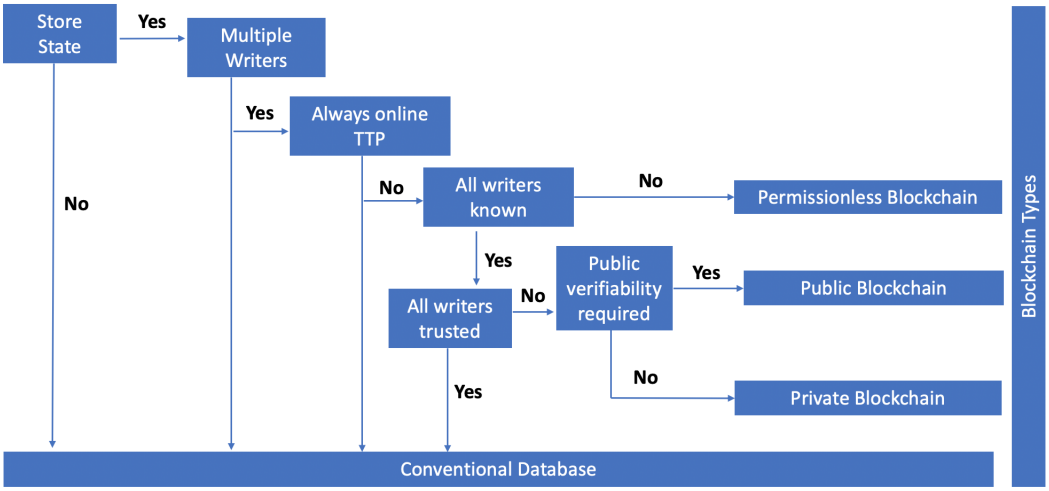

FIGURE 10. Procurement process of blockchain.

for a subset of participants to create their own blockchain network within the existing network without affecting their participation in the network as a whole. For example, two medical centres might share and exchange data between themselves, excluding other medical centres in the same network. It is a variant of private blockchain where participants are organisations. Only a group of organisations can verify or add transactions. Also, the ledger can be open or restricted to selected groups. Although this network model provides the immutable public verification and the high scalability of private networks. Factors, such as cost of ownership and maintenance, might be a burden and difficult to administer going forward.

Most existing literature focuses on how blockchain technology is implemented, rather than assessing its impact and potential applications suitability [106]. Although blockchain solutions have raised interest in implementations for enterprises, limited evolution frameworks and product data can hinder deciding whether blockchain is appropriate for a particular type of deployment as discussed above [107]. One proposed general decision tree framework is illustrated in Figure 10. This flowchart process is based on the existence of a number of basic properties that participate in a blockchain system. As a first evaluation process, if there are no necessities to store system state or any kind of application data, then blockchain usage is not required. Likewise, if there is only one consensus participant that writes or changes the blockchain state, then there is no need to record transition via consensus mechanisms and centralised database will surpass the use of blockchain. The presence of many consensus participants, on the other hand, motivates the need to control the state of updates. This is achievable through a trusted third party (TTP) which is ideally always online to a network of multiple writers or participates. In the unlikely event TTP is not available, then it can operate as a certificate authority in permissioned blockchain model. Depending on public verifiability requirements, either private or public ledger can be used to allow open access to records or restrict to trusted participants. In any blockchain based solutions, it is possible to make use of hashing and encryption to obscure data privacy.

While this framework provides relevant criteria for a single cloud-based applications scenario, multiple cloud deployments would require more specific requirements such as preserving data privacy. Furthermore, the feasibility of blockchain deployment in vendor's cloud environment can be measured through considering the need of data management, data verification, the complexity of existing culture and comparison of blockchain use cases against the intended goals [108].

2) CONSENSUS METHODS AND SMART CONTRACTS OF BLOCKCHAIN

Blockchains have been proposed for various applications such as bitcoin, smart networked systems and cloud systems, without trust approaches between users. Blockchain technology could be an alternative to publicly trusted third- parties. The trustiness of blockchain comes from using consensus methods that validate the data transactions between peer entities [63]. The aim of consensus mechanisms is that the capability of verifying blocks in distributes networks such as those of cloud systems. There are four popular consensus methods: Proof-of-Work (PoW), Proof-of-Stack (PoS), Proof-of-elapsed-Time (PoeT), and Practical byzantine fault tolerance (pBFT): that have been widely employed in blockchain applications. The four methods are discussed separately. In the PoW method, a node in a distributed network can verifiably add a block when it estimates the last block and its nonce by consuming a predefined amount of computational resources. The main target of computing the resources is to avoid Sybil attacks that generate huge numbers of forged identities performing on behalf of one entity [109]. The main challenge of the PoW methods is that a computing system controls more

than half of the total computational resources in a network, which is defined as a 51% attack [110]. The PoW has been deployed in the cryptocurrency bitcoin using an SHA-256 hash function [104].

PoS method was designed by integrating stochastic selection and the influence (i.e., stake) of the participating systems. This method considers an assumption which is computing systems that have a large stake in the blockchain have a significant interest in assuring its integrity. The PoW method has been used in the cryptocurrency applications of BlackCoin or Peercoin [105]. PoeT mechanism is accomplished when a potential validator node demands a protected random waiting time from a reliable execution system that is embedded into a platform, for example, Intel's SGX. Each node waits for the allocated time, and the first to complete it can claim the validation leadership process. Because each reliable computing system in a node has the opportunity of being adopted, the likelihood for any system of being in charge of the leadership process is relative to the number of computing resources that are contributed to the entire network elements [104]. pBFT is a popular permissioned consensus protocol algorithm currently being implemented in Hyperledger Fabric platform [111]. pBFT is commonly used in private blockchain networks, where trust model is implicit between participants compared to PoeT, PoW and PoS consensuses. In addition, it provides an energy efficient consensus protocol to deliver higher transitions throughput without worrying about optimising the platform to large consensus participants.

In blockchain pBFT algorithm is transferred into a group of generals active and passive replications. A primary replica is chosen from the active replicas, which accepts transactions from a client and transfers them to the active replicas for operation. The execution process takes place in four stages, i.e. pre-preparing, planning, agreeing, and responding. In the pre-preparation phase, transactions are primarily sent to all active replicas. Each engaged replica signs the transaction and exchanges it with all the other replicas in the preparation and commit phase., in the reply phase, all the active replicas submit their replies to the primary replica with proof of consensus and the results. Finally, the primary replica gathers all the signed transactions and positions them in a stack [112].

Figure 11 illustrates the method of checking the transaction in a ledger of pBFT. The disadvantage of pBFT is the exponentially increasing message count (complexity of message) relative to PoW, PoeT and, PoS. Furthermore, it operates on the assumption that primary replica does the protocol diligently and does not alter the ordering of transitions. As such this may lead to vulnerabilities and the threat of malicious insider attacks that could break the private network. Nevertheless, since everyone usually knows the identity of the primary, fraudulent behaviours can inevitably be tracked back.

3) SMART CONTRACTS

Smart contracts are fundamentally computer programs used to facilitate, verify and negotiate contract terms between

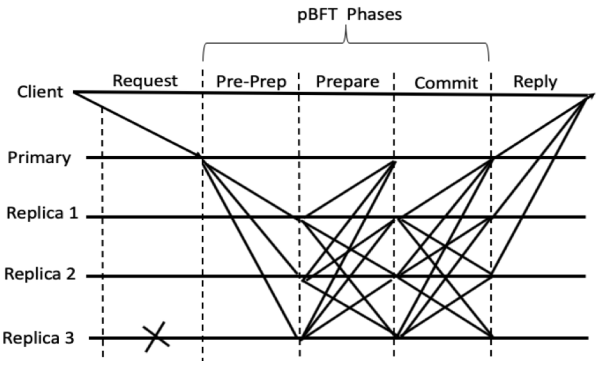

FIGURE 11. pBFT consensus protocol phases.

network participants. They are self-executing contracts between participants and parties that are unable to alter their code, without the need for trusted third-parties [113]. They were originally suggested by Szabo [114], and then they have been used in blockchain as distributed state machines with third parties. Despite the functions of smart contracts are limited due to their small instructions of development, they have been widely applied to cryptocurrency applications, such as Bitcoin, to facilitate a small set of smart contracts. The smart contract has been also used in the Ethereum project to offer a Turing-complete programming language that allows developing arbitrary code on its blockchain [115]. Smart contracts are susceptible to sophisticated attacking techniques such as DoS and DDoS attacks. Some security drawbacks of smart contracts have been addressed using the solidity programming language addressed using a code checker that scans vulnerabilities and exploitable code [116]. In this work, the smart contract concept is utilised to secure the data transaction of live migration in the cloud. This could enhance the process of preserving privacy data transfer in cloud systems.

VII. SECURITY THREATS TO BLOCKCHAIN AND SMART CONTRACTS

The implementation of blockchain-based cloud solutions faces several challenges, including preserving data privacy, operational resilience, trust management, compliance and governance. Software migration from legacy systems to blockchains may also lead to institutional and social inertia, organisations may also expect costs related to tailored implementation and IT infrastructure changes that require third-party support. Therefore a benefit cost analysis can aid in accelerating enterprise wide adoption from such circumstances [117]. Preserving data privacy is a major threat to Blockchain and Smart Contracts, despite the multiple form of implicit security attributes of pseudonymity and tamper-proof architecture. Utilising asymmetric-key cryptography with public and private keys to digitally sign transactions does not guarantee privacy or anonymity, since all transactions and balances are publicly visible [118]. Furthermore, recent studies have demonstrated the feasibility to de-anonymise attacks through studying cryptocurrencies transactional network structures [119], [120].

Meiklejohn *et al.*, were able to link an encrypted transaction to actual individuals by identifying heuristic clusters to classify digital wallet, traders and other previous purchased good and services, since it is easy to label related public keys by communicating with these entities [121]. Similarly, the author in [122] has presented a method to link user public keys to IP addresses where original transactions took place, even when users are behind Network Address Translations (NAT) or proxies. Several approaches to enhance blockchain confidentiality and anonymity have been introduced, which can be classified in two types; mixing services and zero-knowledge proof. The former operates by offering to transfer funds of a user and arbitrarily swap them for funds of other users to conceal their ownership, though these are not shielded from the service's theft. Whereas the latter, uses cryptographic accumulator to validate transactions with digital signature from a pool of valid funds that users are able to exchange random funds. However, this method introduces computational overhead and reveals transaction amounts. Sasson [123] was designed to counter these problems and further improve transactions anonymity.

Although private and consortium blockchains offer better privacy protection and less vulnerability to malicious attack, their propensity to be centralised within a single CSP also makes them less efficient and prone to single point of failure [124]. On the other hand, public, distributed, and decentralised ledger draws lack of trust between parties. Thus, CSP are usually reluctant to share data or report intrusion events between each other because of concerns around data confidentiality and privacy. It is quite difficult to measure the level of reputation between untrusted participants. Another major threat is insider attacks such as collusion and betrayal attacks [14], where malicious nodes collaborate to give false information and degrade the efficiency of blockchain operations.

The ability to maintain data storage with regulatory compliance can be costly and time-consuming [117]. Organisations are increasingly using CSP for scalable storage of various data sets where security and regulatory constraints are still a major threat. Moreover, blockchains and smart contracts operate through different jurisdictions, making it difficult to guarantee that all rules within compliance. Consequently, a decentralised architecture should enable compliance with regulations and ensures control of cost and policy with regards to the CSP. Despite the inherent cyber resilience capabilities of the blockchain and smart contract-based systems, they are not immune against all forms of malicious adversaries and emerging cyber security threats. Understanding such threats is crucial for their wide adaptation or potential offering of blockchain-as-a-service (BcaaS) towards decentralised ecosystem and achieving Web 3.0 technologies [125], [126]. Blockchain and smart contact threats can be classified into five board categories that could compromise systems built on it as illustrated in the Table 4. It includes, security threats and their attack vectors on blockchain-based networks, transaction verification mechanisms (TVM), user wallet, mining pools and smart contract-based systems.

The Blockchain network-related attacks include BGP attacks, DDoS, Eclipse, Sybil and time-jacking among others. The attacker's aim for each of these attacks is to separate users and miners from the actual network, block their access to network resources, or establish network partitioning, and implement contradictory peer laws. The distributed denial-of-service (DDoS) attack is one of the most frequent and damaging threat to any online cloud service [140]. Both blockchain and smart contracts are still vulnerable to DDoS attacks, despite having a P2P network architecture. These attacks have often occurred in applications such as Vasek *et al.* [141]. While targeting a blockchain network, attackers intend to break down a node by using numerous requests to absorb all of its computing power leading to disruption in network's mining pools, digital wallets and smart contracts services [142]. It can be costly to initiate such an attack on public blockchains, whereas in pBFT based private networks the malicious adversary only need to own 33 percent of active replicas to lunch a successful attack [143]. Another significant external threat to that can impact individual node or the whole blockchain network is routing attacks such as DNS or BGP hijacking. This attack can occur as a result of router manipulation through mis-configurations or malicious intent [144].

Apostolaki *et al.* [127] presented a comprehensive taxonomy of BGP attacks and their impact targeting single nodes and the whole blockchain network with large scale attacks. Through isolating part of the network, malicious miners would exploit vulnerabilities in these protocols to intercept and reroute blockchain network traffic and eventually delay block propagation, waste considerable mining power and allow for a number of exploits, such as double expenditure. In a similar manner, the eclipse attack operates to isolating neighbouring nodes from real network by allowing intruders to monopolise all incoming and outgoing communications between the target and the other network participants [2]. This allows attackers to pollute the victim's view of the network and invoke various forms of selfish mining and double speeding attacks. Sybil attack, on the other hand, seeks to attack the whole P2P network by creating large numbers of forged pseudonymous identities to achieve illegitimate network influence [109]. Finally, the time-jacking attack exploits a flaw in the processing of Bitcoin timestamps. An attacker adjusts the node's network time counter during a time-jacking attack and forces the node to accept an alternate blockchain [145]. This can be accomplished by adding multiple fake peers with incorrect timestamps to the network by a malicious client. This attack, however, can be mitigated by limiting the time ranges of acceptance or using time synchronisation of the node [131].

The majority or also known as 51% attack can occur theoretically when a group of malicious miners are able to exploit and control the majority of the total network's hash

TABLE 4. Threats of blockchains and smart contracts.

Security Threats	Attack Type	Primary target	Countermeasures
Blockchain-based Network Threats	DNS/BGP hijacking	Blockchain network, miners	Awareness of routing [127]
	DoS/DDoS	Blockchain network, applications, miners	Increasing throughput and block size [128]
	Eclipse	Miners, users	Whitelists [129]
	Sybil	Blockchain network, miners	Two-party mixing protocol [130]
	Time-jacking	Miners, applications, users	Time synchronisation [131]
TVS/Double Spending Threats	Majority (51%)	Blockchain network, miners, applications	Using two phase proof of work [132]
	Finney	Miners, mining pool, users	Multiple confirmations of transactions
	Withholding	Miners, mining pool	Only trusted miners in mining pool,,cryptographic commitment schemes [133]
User Wallet Threats	Wallet theft	Miners, users' private keys	Hardware tokens [134], enforce threshold signatures [135]
Mining Pools Threats	Selfish Mining	Blockchain network, miners, mining pool	Timestamp-free technique [136]
Smart Contact-based Threats	Re-entrancy	Smart contract, applications, users	Symbolic execution [137]
	Integer overflow	Applications, users	Code checkers, security patching [138]
	Short address	Applications, users	Security patching, checksum addresses [112]
	Transaction ordering dependence	Smart contract, applications, users	Security patching [139]

rate or computational power. This attack demonstrates how blockchains are not protected from all cyber attacks. The majority attack is a well-known vulnerability in blockchain platform that enables attackers to append their fraudulent blocks in the blockchain network with high probability and breaching the chain's security [146]. Consequently, granting attackers the power to destroy the integrity of the entire network through acts such as double spending, remove, change and reverse transactions, create an alternative fork to split the blockchain and prevent some or all new transactions form gaining any confirmations during the time they are in control [112]. Another variant of double spending attack, includes Finney and Withholding attacks that deliberately cover, fake or preserve important information that must be transmitted throughout the network. The former only occurs if a trader confirms the transaction only once. The attacker produces a transaction, calculates a block and does not choose to transmit the block. Meanwhile, generates a duplicate of previous transaction then releasing it to public network. Similarly, the withholding attacks is conducted against distributed mining pools in an attempt to harm the pool owner by withdrawing a legitimate PoW [112].

Blockchain-based systems utilise private key-based encryption, However, passwords are still most common form of client authentication [147]. Different types of key management software are used to store these keys in an electronic wallet. E-wallet theft is primarily done through techniques that include device manipulation (bugs & malware), faulty app setup and inappropriate wallet usage. Another attack that pose serious risks to the stability of blockchain network is attacks on mining pools. Miners also merge their computer power to build a mining pool such as BTC and Slushpool. This allows them to collect additional blocks and receive a portion of the reward. However, pools are susceptible to selfish mining attacks [148], where malicious miners would use a tactic to purposely keep their blocks private in order to enhance their rewards. Instead of publishing their own blocks back to the rest of the network, they continue to mine their own private blocks to get a large chain than the public. Consequently, a selfish miner would gain competitive advantage by increasing their revenue rewards and wasting honest miners computing power and capital.

The most popular smart contract application is Ethereum, which leverages Solidity for contract development, others EOS, and NEO platforms. Solidity is an object-oriented programming language based on JavaScript, C++ and Python and written contract are compiled to Ethereum Virtual Machine (EVM) bytecode [148]. Vulnerabilities in smart contracts is not immune from cyber attacks, particularly in the cloud domain and suffer from similar attack vectors that threaten the blockchain technologies. Some of the security issues associated with smart contracts that leads to a series of attacks come from possible bugs and malware in source code that targets EVM and blockchain network. Potential attacks as a consequence of these security breaches include the re-entrancy attack, where it occurs when a developer creates a function that externally calls another untrusted contract and then invoke a malicious code such as multiple currency withdrawals and losing the entire balance in the contract; Overflow attack, occurs when if a number increases above its maximum value, for example solidity can manage up to 256-bit numbers, increasing the value by 1 would trigger

an overflow; short address attack, occurs when a contract receives fewer data that expected by exploiting a bug in solidity to auto-pad missing bytes with extra zeros gaining extra tokens; Transaction-Ordering Dependence, where transactions within the blockchain network are not ordered or executed correctly which makes them subject to manipulation [112], [149]. Failure to minimise such issues may result in substantial financial loss or data compromise and service availability.

VIII. CURRENT RESEARCH IN CLOUD COMPUTING, IDS, BLOCKCHAIN AND SMART CONTRACTS

This section outlines the current state of research combining the areas of cloud computing, IDS, and the emerging areas of blockchain and smart contracts. It addresses the current work, and also current weaknesses and deficiencies in current cloud vendor deployments. These are considered research opportunities.

A. CURRENT RESEARCH IN CLOUD, IDS AND BLOCKCHAIN

Due to the heterogeneous nature and virtualisation of cloud computing environments, determining a collective IDS structure is challenging. Multiple researchers tackled this problem by developing more effective model designs at the application, platform, and infrastructure layers, yet separately [150]. Examples include the work by Gustavo and Miguel which analysed anomaly-based intrusion detection with data acquired from production environments hosting a SaaS web application of large dimensions [151]. The work shows that detection of attacks at the application layer is feasible, with the n-gram model provides the least false positives and high detection rate. Nevertheless, the work did not consider an effective approach for deploying the system in a real cloud computing environment.

In another work that deals with attacks in Infrastructure as a Service cloud, Tupakula *et al.* [152] considered the design choices and countermeasures for securing customer virtual machines in the cloud. The proposed model handles network traffic from each virtual machine even if multiple virtual machines are sharing a single IP address. However, this model cannot protect the system if the infrastructure collapsed due to the higher risk attacks over the system. In order to tackle this issue, Wang *et al.* [153] developed a collaborative IDS with a centralised management method to deliver faster and accurate detection. But, this system might not scale well and the performance might decrease as the data load build-up in the central node, representing a single point of failure that remains unstable in the cloud. Others followed a machine learning approach, as in the work of Vieira *et al.* [154], where an Artificial Neural Network (ANN) algorithm was employed to train and validate the proposed IDS with a designed prototype using a Grid-M middleware.

The grid and cloud Computing IDS demonstrated an improved computational power, as each node was processed individually; however, the difference between the grid and cloud system in terms of security policies, business modes, and systems requirements requires specific IDS design for cloud and grid to be executed separately [155]. In a similar work that employs an ANN algorithm in an IDS structure for each user of cloud computing services, a single controller was used to manipulate the IDS instances relying on the knowledge base using pattern matching of multiple false login attempts. This machine learning structure has several limitations related to the lack of sensitivity and scalability of central manager failure. Conversely, Kholidy and Baiardi [156] proposed a system without a central manager coordinator. The distributed system has a P2P network architecture with a hybrid detection capability of host and network data, which integrates well for cloud computing. Tan *et al.* [27] proposed also a collaborative IDS which associates malicious events between different IDSs to enhance the IDS efficiency.

The hybrid of HIDSs and NIDSs that the system provides facilitates the implementation of signature- and anomaly-based detection at the host and network platforms. The dual modes of the system, namely the cooperative agent and central coordinator, facilitates for implementing several security mechanisms. Although these collaborative systems are claimed to be scalable, they cannot efficiently detect large-scale distributed anomalies, and there is no central correlation handler to merge all the alert information reliably to discover intrusions. Other researchers applied an ontological approach for cyber security operational information based on actual biosecurity operations, where Takahashi *et al.* [157] produced an ontological IDS for cloud computing using a scoring-based system for detecting vulnerabilities. The system identifies data asset decoupling, the composition of multiple resources and external resource usage, and applied all together as a set of public cyber security terms in cloud computing environments.

Lee *et al.* [158] proposed a multi-stage IDS supported by log management. The system relies on different security levels to constrain access rights in cloud systems, where attack events are issued by properly weighting each security risk. However, these IDSs are prone to zero-day attacks, and usually, require a prolonged processing time in the large size networks of the cloud computing environments. In this context, Shelke *et al.* [25] dealt with the issue of handling large scale network traffic and associated administrative control of data and application in a cloud computing environment by a multi-threaded distributed IDS technique. The main application of the technique was for detecting Distributed Denial of Service (DDoS) and Cross-Site Scripting (XSS) attacks. It has the advantage of processing large flow of data packets, and can generate reports by incorporating knowledge and behavior analysis to detect intrusions. Similarly, Zarrabi and Zarrabi [159] suggested a host and network IDS for identifying suspicious activities in cloud computing. The cloud intrusion detection system service was deployed as a SaaS application for protecting overcoming cyber attacks. Also, Alharkan and Martin [160] proposed IDS as a service for cloud systems to detect attacks. The implementation was at

the infrastructure level of a public cloud (IaaS) by providing a detection technology, which is highly scalable by the cloud users.

For works that applied a hybrid approach, Rajendran *et al.* [161] proposed a hybrid IDS which could detect different attacks in cloud environments. It was tested against common attacks, but the results showed that this system cannot detect zero-day attacks efficiently. Nikolai and Wang [162] suggested a hypervisor-based cloud IDS which does not demand additional software installed in VMs. Additionally, Mehmood *et al.* [163] employed mobile agents to identify distributed attacks in the cloud. The main objective was to recognise the intrusions on VMs, detect vulnerable ports, and correlate suspicious activities events to discover distributed intrusions in a cloud-based network. Attack alerts were also carried using mobile agents from user VMs to the management server where correlation happens. In the same manner, Vieira *et al.* [164] proposed an IDS using big data tools for data analytics and expected utility function for decision making. The work aims to mitigate intrusions that break, confidentiality, integrity, and availability in cloud computing platforms. Taking into consideration the properties of self-awareness, self-optimisation and self-healing in design an effective IDS. However, these IDSs are not scalable and robust enough to recognise distributed attacks as each IDS operates independently, and lack the capability of detecting new attacks effectively.

The blockchain and IDS solutions have been used in a few studies to provide trusted collaborative IDSs in network and cloud systems. For instance, Alexopoulos *et al.* [165] surveyed the methods of integrating CIDSs and blockchains. Especially, the authors introduced the concept of using blockchain techniques for enhancing the credibility of CIDSs. It is noted that characteristics of blockchain can benefit CIDSs in the ways of trusting each IDS and offering accountability and consensus methods. In [166], the authors also reviewed the significance of using blockchain and its theoretical approaches that would be employed to secure CIDSs. Liang *et al.* [167] proposed a decentralised and secured data provenance framework that offer tamper-proof data blocks. This framework allows data accountability and improves data privacy and could prevent inference attacks from exploiting cloud systems. To sum up, the integration of blockchain and CIDS solutions would considerably improve security levels when they are deployed in cloud systems.

IX. CHALLENGES AND FUTURE DIRECTIONS OF IDS AND BLOCKCHAIN IN THE CLOUD

Although the cyber security research community has developed several intrusion detection techniques and distributed frameworks that are capable of protecting large scale networks, they still face potential cloud-specific challenges. These challenges which originated from cloud computing environments are described as follows [63], [151], [152], [156], [158], [168];

- The exponential growth of recent zero-day attacks and their vulnerabilities makes the ability to detect sophisticated insider and outsider attacks from network and cloud systems complex.
- Providing continuous and active monitoring to remediate possible incidents and intrusions in real-time cloud suffer from several limitations.
- Developing a self-adaptation capability to optimise and significantly reduce the intervention of operators is a significant challenge.
- Designing an adaptive IDS architecture to handle large scale dynamic autonomous computing is complicated.
- Intrusion systems must be resilient to failure and compromise by protecting itself from insider and outsider unauthorised access or attacks.
- Cloud Service Providers (CSPs) are incompetence of synchronisation of risk profile.
- Access to publicly available and open-source IDS datasets that could be used to validate detection models is lacking particularly for a cloud-based domain.
- The integration of IDS and blockchain technologies is still complex due to the deployment difficulty and high computational resources needed.

The ability to detect sophisticated insider and outsider attacks from network and cloud systems with a minimum number of FP and FN alarm rates is very complex, due to the exponential growth of recent zero-day attacks and their vulnerabilities. Also, building and maintaining a profile that incorporates all possible legitimate behaviours is challenging as the boundary between normal and abnormal behaviour is often inaccurate. For example, legitimate events that fall close to the abnormal region is detected as attacks and vice versa. IDS should learn and improve its detection capability while maintaining efficient use of cloud resources and reduce location dependency. Hence, developing IDS independent of the deployed IT environment with the desired level of performance and security is a challenge [63], [151].

Providing continuous and active monitoring to remediate possible incidents and intrusions in real-time cloud suffer from several limitations as such when dealing with noisy data and malicious encrypted packets. The former limitation deemed as noisy or bad packets could be generated from software bugs, corrupt DNS data, or local packets that escaped which can create significantly higher false alarms, while the latter can allow for an intrusion to the network via the encrypted packet that is undiscovered until more significant network intrusions have occurred. Furthermore, to maintain and deliver the quality of service, a cloud IDS with several administrators should have minimised or no human interference to evade wasting time for administration responses [152].

Developing a self-adaptation capability to optimise and significantly reduce the intervention of operators is a significant challenge. It is essential to design an automated and adaptive cloud-based IDS that is capable of adapting

to changing requirements such as the size of computing resources, environment configurations, and deployment locations. Additionally, this enables effective, monitoring, analysing and managing IDS alerts across distributed infrastructure.

Designing an adaptive IDS architecture to handle large scale dynamic autonomous computing can be complicated. IDSs should be scalable to efficiently handle a large number of VMs available in the cloud, their communication and computational load. It must scale dynamically to the new addition of VMs to fit into the extended cloud network. It is critical to adopt this capability to preserving these changes to their threshold, which is the baseline between normal and attacks events. Intrusion systems must be resilient to failure and compromise by protecting itself from insider and outsider unauthorised access or attacks. It should be capable of authenticating network devices and IDS mutually, protecting its data, and preventing any loopholes, which may create new vulnerabilities.

Cloud Service Providers (CSPs) are incompetence of synchronisation of risk profile. The lack of visibility and transparency to share logs, incidents, and vulnerabilities between different CSPs leads to the computational complexities of data pre-processing in the training phase. While each system runs and detects malicious events independently, their information and activities must be synchronised for discovering distributed and concurrent intrusions. Furthermore, tracking migrating data across different platforms and multi CSPs while maintaining a degree of trust and privacy is also a challenging task.

Access to publicly available and open-source IDS datasets that could be used to validate detection models is lacking particularly for a cloud-based domain. Most of the existing datasets often lack attack diversity, inaccurate labelling, and incomplete capture traffic packet content including both header and payload. Also, fog and edge computing architectures demand different CIDS approaches to efficiently identity unknown malicious events in real-time.

The integration of IDS and blockchain technologies is still complex due to the deployment difficulty and high computational resources needed. There are various distributed IDSs that should be integrated with a database management system that allows generating alerts in real-time. Moreover, the adaptation of blockchain technology, especially during the live migration process, and the configuration of CIDSs and their database, still needs more investigation to ensure the high credibility of IDSs and preserving data privacy and to verify data transmission and cloud nodes [69], [104], [105], [109]. In summary, there are numerous technological and research challenges in this space.

A. CONCLUSION

This paper has explained the background of intrusion detection, blockchain and cloud computing systems. Moreover, the previous studies related to these systems have also been described. The challenges of using blockchain and intrusion detection systems have also been examined and analyzed. This work has found that the growing usage of cloud in cyber security is significant, but there is still a significant need for further research in this field. Specifically, the live migration process of cloud systems represents a potential threat, that exposes organizations making use of cloud features. In this space, there is an urgent requirement for technologies that can discover insider and outsider attacks whilst maintaining data privacy. Blockchain looks to be a promising technology that would be implemented in solutions in this space to trust cloud nodes and intrusion detection engines.

Cloud is increasing in importance, but also poses new and unique security challenges. Several of these relate to the fact that cloud implementations are at large scale, and have challenges in both maintaining sovereignty and also allowing full client access, are highly connected and hence vulnerable to different attacks, and are built upon multiple technologies, that may themselves be vulnerable or exploited. Additionally, cloud systems are a multi-vendor environment, and virtual machines (VMs) can be migrated live between systems. This provides a motivated, sophisticated attacker opportunity to corrupt, disrupt or collect information on the VM in transit. Finally, intrusion detection systems are in a state of immense change to operate in the fast-paced area of cloud security. Technologies such as blockchain and smart contracts are being leveraged to provide benefit to these environments, but there is still opportunity to work further in this area.

REFERENCES

[1] R. Huang. (2018). *New Open-Sourced Innovation Aims to Reduce the Risk of 51% Attacks*. Accessed: Apr. 9, 2020. [Online]. Available: https://www.forbes.com/sites/rogerhuang/2018/10/11/new-open-sourced-inn%ovation-aims-to-reduce-the-risk-of-51-attacks/#d10f98152088

[2] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, vol. 107, pp. 841–853, Jun. 2020.

[3] G. W. Peters and E. Panayi, "Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the Internet of money," in *Banking Beyond Banks and Money*. Cham, Switzerland: Springer, 2016, pp. 239–278.

[4] M. Keshk, B. Turnbull, N. Moustafa, D. Vatsalan, and K.-K.-R. Choo, "A privacy-preserving-framework-based blockchain and deep learning for protecting smart power networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5110–5118, Aug. 2020.

[5] N. Moustafa, B. Turnbull, and K.-K.-R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4815–4830, Jun. 2019.

[6] F. Tian, "A supply chain traceability system for food safety based on HACCP, blockchain & Internet of Things," in *Proc. Int. Conf. Service Syst. Service Manage.*, Jun. 2017, pp. 1–6.

[7] M. Keshk, N. Moustafa, E. Sitnikova, and G. Creech, "Privacy preservation intrusion detection technique for SCADA systems," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2017, pp. 1–6.

[8] N. Moustafa, G. Creech, E. Sitnikova, and M. Keshk, "Collaborative anomaly detection framework for handling big data of cloud computing," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2017, pp. 1–6.

[9] N. Moustafa, G. Creech, and J. Slay, "Anomaly detection system using beta mixture models and outlier detection," in *Progress in Computing, Analytics and Networking*. Singapore: Springer, 2018, pp. 125–135.

[10] T. Marsden, N. Moustafa, E. Sitnikova, and G. Creech, "Probability risk identification based intrusion detection system for SCADA systems," in *Proc. Int. Conf. Mobile Netw. Manage.* Cham, Switzerland: Springer, 2017, pp. 353–363.

[11] N. Moustaf and J. Slay, "Creating novel features to anomaly network detection using DARPA-2009 data set," in *Proc. 4th Eur. Conf. Cyber Warfare Secur. Academic Conf. Limited*, 2015, pp. 204–212.
[12] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu, and I. Khalil, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems," *IEEE Trans. Sustain. Comput.*, early access, 2019, doi: 10.1109/TSUSC.2019.2906657.
[13] N. Moustafa, K.-K.-R. Choo, I. Radwan, and S. Camtepe, "Outlier Dirichlet mixture mechanism: Adversarial statistical learning for anomaly detection in the fog," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 8, pp. 1975–1987, Aug. 2019.
[14] W. Li, W. Meng, L.-F. Kwok, and H. H. S. Ip, "Enhancing collaborative intrusion detection networks against insider attacks using supervised intrusion sensitivity-based trust management model," *J. Netw. Comput. Appl.*, vol. 77, pp. 135–145, Jan. 2017.
[15] M. Keshk, N. Moustafa, E. Sitnikova, and B. Turnbull, "Privacy-preserving big data analytics for cyber-physical systems," *Wireless Netw.*, vol. 24, pp. 1–9, Dec. 2018.
[16] O. AlKadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "Mixture localization-based outliers models for securing data migration in cloud centers," *IEEE Access*, vol. 7, pp. 114607–114618, 2019.
[17] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017.
[18] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019.
[19] S. Sharma and A. Kaul, "A survey on intrusion detection systems and honeypot based proactive security mechanisms in VANETs and VANET cloud," *Veh. Commun.*, vol. 12, pp. 138–164, Apr. 2018.
[20] O. S. Alkadi, N. Moustafa, and B. Turnbull, "A collaborative intrusion detection system using deep blockchain framework for securing cloud networks," in *Proc. SAI Intell. Syst. Conf.* Amsterdam, The Netherlands: Springer, 2020.
[21] J. Xie, H. Tang, T. Huang, F. R. Yu, R. Xie, J. Liu, and Y. Liu, "A survey of blockchain technology applied to smart cities: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2794–2830, 3rd Quart., 2019.
[22] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, and K. Zheng, "Survey on blockchain for Internet of Things," *Comput. Commun.*, vol. 136, pp. 10–29, 2019.
[23] S. Xie, Z. Zheng, W. Chen, J. Wu, H.-N. Dai, and M. Imran, "Blockchain for cloud exchange: A survey," *Comput. Electr. Eng.*, vol. 81, Jan. 2020, Art. no. 106526.
[24] J. Bond, *The Enterprise Cloud: Best Practices for Transforming Legacy IT*. Newton, MA, USA: O'Reilly Media, 2015.
[25] M. P. K. Shelke, M. S. Sontakke, and A. D. Gawande, "Intrusion detection system for cloud computing," *Int. J. Sci. Technol. Res.*, vol. 1, no. 4, pp. 67–71, 2012.
[26] P. M. Mell and T. Grance, "The NIST definition of cloud computing," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. SP 800-145, 2011.
[27] Z. Tan, U. T. Nagar, X. He, P. Nanda, R. P. Liu, S. Wang, and J. Hu, "Enhancing big data security with collaborative intrusion detection," *IEEE Cloud Comput.*, vol. 1, no. 3, pp. 27–33, Sep. 2014.
[28] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Gener. Comput. Syst.*, vol. 28, no. 3, pp. 583–592, Mar. 2012.
[29] Z. Kozhirbayev and R. O. Sinnott, "A performance comparison of container-based technologies for the cloud," *Future Gener. Comput. Syst.*, vol. 68, pp. 175–182, Mar. 2017.
[30] C. Pahl, "Containerization and the PaaS cloud," *IEEE Cloud Comput.*, vol. 2, no. 3, pp. 24–31, May 2015.
[31] A. Martin, S. Raponi, T. Combe, and R. Di Pietro, "Docker ecosystem–vulnerability analysis," *Comput. Commun.*, vol. 122, pp. 30–43, Jun. 2018.
[32] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.
[33] E. Spangler, "Method and system for measurement of road profile," U.S. Patent 4 741 207, May 3, 1988.
[34] S. Singh and I. Chana, "A survey on resource scheduling in cloud computing: Issues and challenges," *J. Grid Comput.*, vol. 14, no. 2, pp. 217–264, Jun. 2016.
[35] D. He, Z. Wang, and J. Liu, "A survey to predict the trend of ai-able server evolution in the cloud," *IEEE Access*, vol. 6, pp. 10591–10602, 2018.
[36] T. Halabi and M. Bellaiche, "Towards security-based formation of cloud federations: A game theoretical approach," *IEEE Trans. Cloud Comput.*, early access, 2018, doi: 10.1109/TCC.2018.2820715.
[37] E. Smith and M. Shirer. (2019). *Worldwide Public Cloud Services Spending Forecast to Reach 210 Billion This Year, According to IDC*. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS44891519
[38] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic resource management using virtual machine migrations," *IEEE Commun. Mag.*, vol. 50, no. 9, pp. 34–40, Sep. 2012.
[39] Y. Koh, R. Knauerhase, P. Brett, M. Bowman, Z. Wen, and C. Pu, "An analysis of performance interference effects in virtual environments," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, Apr. 2007, pp. 200–209.
[40] L. Rodero-Merino, L. M. Vaquero, E. Caron, A. Muresan, and F. Desprez, "Building safe PaaS clouds: A survey on security in multitenant software platforms," *Comput. Secur.*, vol. 31, no. 1, pp. 96–108, Feb. 2012.
[41] D. Basu, X. Wang, Y. Hong, H. Chen, and S. Bressan, "Learn-as-you-go with Megh: Efficient live migration of virtual machines," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 1786–1801.
[42] L. Fujitsu. (2011). *White Paper Virtualization at Fujitsu*. Accessed: Jan. 25, 2020. [Online]. Available: https://www.fujitsu.com/th/th/Images/wp-virtualization-at-fujitsu-ww-en%.pdf
[43] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proc. 2nd Conf. Symp. Netw. Syst. Design Implement.*, vol. 2, 2005, pp. 273–286.
[44] H. Liu and B. He, "VMbuddies: Coordinating live migration of multi-tier applications in cloud environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 1192–1205, Apr. 2015.
[45] N. Tziritas, T. Loukopoulos, S. U. Khan, C.-Z. Xu, and A. Y. Zomaya, "Online live VM migration algorithms to minimize total migration time and downtime," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2019, pp. 406–417.
[46] F. Xu, F. Liu, L. Liu, H. Jin, B. Li, and B. Li, "IAware: Making live migration of virtual machines interference-aware in the cloud," *IEEE Trans. Comput.*, vol. 63, no. 12, pp. 3012–3025, Dec. 2014.
[47] R. Shaikh and M. Sasikumar, "Data classification for achieving security in cloud computing," *Procedia Comput. Sci.*, vol. 45, pp. 493–498, Jan. 2015.
[48] S. Shastri and D. Irwin, "HotSpot: Automated server hopping in cloud spot markets," in *Proc. Symp. Cloud Comput.*, Sep. 2017, pp. 493–505.
[49] S. Singh, Y.-S. Jeong, and J. H. Park, "A survey on cloud computing security: Issues, threats, and solutions," *J. Netw. Comput. Appl.*, vol. 75, pp. 200–222, Nov. 2016.
[50] Z. Cao, J. Lin, C. Wan, Y. Song, Y. Zhang, and X. Wang, "Optimal cloud computing resource allocation for demand side management in smart grid," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1943–1955, Jul. 2017.
[51] D. Merkel, "Docker: Lightweight Linux containers for consistent development and deployment," *Linux J.*, vol. 2014, no. 239, p. 2, 2014.
[52] (2019). *Flockport(LXC)*. Accessed: Apr. 9, 2020. [Online]. Available: http://www.flockport.com
[53] C. Pahl, A. Brogi, J. Soldani, and P. Jamshidi, "Cloud container technologies: A state-of-the-art review," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 677–692, Jul. 2019.
[54] S. Soltesz, H. Pötzl, M. E. Fiuczynski, A. Bavier, and L. Peterson, "Container-based operating system virtualization: A scalable, high-performance alternative to hypervisors," in *Proc. 2nd ACM SIGOPS/EuroSys Eur. Conf. Comput. Syst.*, 2007, pp. 275–287.
[55] T. Combe, A. Martin, and R. Di Pietro, "To docker or not to docker: A security perspective," *IEEE Cloud Comput.*, vol. 3, no. 5, pp. 54–62, Sep. 2016.
[56] J. Chelladhurai, P. R. Chelliah, and S. A. Kumar, "Securing docker containers from denial of service (DoS) attacks," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun. 2016, pp. 856–859.
[57] D. Yu, Y. Jin, Y. Zhang, and X. Zheng, "A survey on security issues in services communication of microservices-enabled fog applications," *Concurrency Comput., Pract. Exp.*, vol. 31, no. 22, p. e4436, Nov. 2019.
[58] D. Bernstein, "Containers and cloud: From LXC to docker to kubernetes," *IEEE Cloud Comput.*, vol. 1, no. 3, pp. 81–84, Sep. 2014.
[59] R. Shu, X. Gu, and W. Enck, "A study of security vulnerabilities on docker hub," in *Proc. 7th ACM Conf. Data Appl. Secur. Privacy*, Mar. 2017, pp. 269–280.

[60] J. Watada, A. Roy, R. Kadikar, H. Pham, and B. Xu, "Emerging trends, techniques and open issues of containerization: A review," *IEEE Access*, vol. 7, pp. 152443–152472, 2019.
[61] N. Kajal, N. Ikram, and Prachi, "Security threats in cloud computing," in *Proc. Int. Conf. Comput., Commun. Autom.*, May 2015, pp. 214–219.
[62] W. A. Conklin, G. White, C. Cothren, R. Davis, and D. Williams, *Principles of Computer Security*. New York, NY, USA: McGraw-Hill, 2015.
[63] A. Patel, M. Taghavi, K. Bakhtiyari, and J. Celestino, Jr., "An intrusion detection and prevention system in cloud computing: A systematic review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 25–41, 2013.
[64] *The Treacherous Twelve-Cloud Computing Top Threats in 2016*, Cloud Security Alliance, Seattle, WA, USA, 2016.
[65] J. Williams and D. Wichers, "The ten most critical Web application security risks," OWASP Found., Bel Air, MD, USA, Tech. Rep. 1, 2017.
[66] Verizon. (2019). *2019 Data Breach Investigations Report*. Accessed: Jan. 25, 2020. [Online]. Available: https://enterprise.verizon.com/resources/executivebriefs/2019-dbir-exec%utive-brief.pdf
[67] McAfee Labs. (2019). *McAfee Labs Threats Report December 2019*. Accessed: Jan. 25, 2020. [Online]. Available: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-thr%eats-aug-2019.pdf
[68] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42–57, Jan. 2013.
[69] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," in *Proc. Conf. Theory Appl. Cryptogr.* Berlin, Germany: Springer, 1990, pp. 437–455.
[70] C. Williams. (2019). *Bezos DDoS'd: Amazon Web Services' DNS Systems Knackered by Hours-Long Cyber-Attack*. Accessed: Jan. 25, 2020. [Online]. Available: https://www.theregister.co.uk/2019/10/22/aws_dns_ddos/
[71] I. Homoliak, F. Toffalini, J. Guarnizo, Y. Elovici, and M. Ochoa, "Insight into insiders and it: A survey of insider threat taxonomies, analysis, modeling, and countermeasures," *ACM Comput. Surv.*, vol. 52, no. 2, p. 30, 2019.
[72] N. Elmrabit, S.-H. Yang, and L. Yang, "Insider threats in information security categories and approaches," in *Proc. 21st Int. Conf. Autom. Comput. (ICAC)*, Sep. 2015, pp. 1–6.
[73] Y. Nugraha, I. Brown, and A. S. Sastrosubroto, "An adaptive wideband Delphi method to study state cyber-defence requirements," *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 1, pp. 47–59, Jan. 2016.
[74] A. J. Duncan, S. Creese, and M. Goldsmith, "Insider attacks in cloud computing," in *Proc. IEEE 11th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Jun. 2012, pp. 857–862.
[75] A. A. Cárdenas, P. K. Manadhata, and S. P. Rajan, "Big data analytics for security," *IEEE Security Privacy*, vol. 11, no. 6, pp. 74–76, Nov./Dec. 2013.
[76] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and big heterogeneous data: A survey," *J. Big Data*, vol. 2, no. 1, p. 3, Dec. 2015.
[77] K. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems (IDPS)," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 1, 2012.
[78] D. Moon, S. B. Pan, and I. Kim, "Host-based intrusion detection system for secure human-centric computing," *J. Supercomput.*, vol. 72, no. 7, pp. 2520–2536, Jul. 2016.
[79] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019.
[80] S. Pontarelli, G. Bianchi, and S. Teofili, "Traffic-aware design of a high-speed FPGA network intrusion detection system," *IEEE Trans. Comput.*, vol. 62, no. 11, pp. 2322–2334, Nov. 2013.
[81] J. Peng, K.-K.-R. Choo, and H. Ashman, "User profiling in intrusion detection: A review," *J. Netw. Comput. Appl.*, vol. 72, pp. 14–27, Sep. 2016.
[82] M. Kumar, M. Hanumanthappa, and T. S. Kumar, "Encrypted traffic and IPsec challenges for intrusion detection system," in *Proc. Int. Conf. Adv. Comput.* New Delhi, India: Springer, 2013, pp. 721–727.
[83] T. Garfinkel and M. Rosenblum, "A virtual machine introspection based architecture for intrusion detection," in *Proc. NDSS*, vol. 3, 2003, pp. 191–206.
[84] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Taxonomy and survey of collaborative intrusion detection," *ACM Comput. Surv.*, vol. 47, no. 4, p. 55, Jul. 2015.
[85] C. Fung and R. Boutaba, *Intrusion Detection Networks: A Key to Collaborative Security*. Boca Raton, FL, USA: CRC Press, 2013.
[86] S. R. Snapp, J. Brentano, G. V. Dias, T. L. Goan, L. T. Heberlein, C.-L. Ho, K. N. Levitt, B. Mukherjee, S. E. Smaha, and T. Grance, *DIDS (Distributed Intrusion Detection System)—Motivation, Architecture, and an Early Prototype*. Reading, MA, USA: Addison-Wesley, 1997, pp. 211–227.
[87] V. Yegneswaran, P. Barford, and S. Jha, "Global intrusion detection in the domino overlay system," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1, 2003.
[88] R. Huebsch, B. Chun, J. M. Hellerstein, B. T. Loo, P. Maniatis, T. Roscoe, S. Shenker, I. Stoica, and A. R. Yumerefendi, "The architecture of pier: An Internet-scale query processor," in *Proc. CIDR*, 2005, pp. 28–43.
[89] B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," in *Proc. 8th IEEE Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jun. 2016, pp. 581–585.
[90] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *Proc. 2nd Int. Conf. Adv. Cloud Big Data*, Nov. 2014, pp. 247–252.
[91] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. 9th EAI Int. Conf. Bio-Inspired Inf. Commun. Technol. (Formerly BIONETICS)*, 2016, pp. 21–26.
[92] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
[93] M. Abdelsalam, R. Krishnan, Y. Huang, and R. Sandhu, "Malware detection in cloud infrastructures using convolutional neural networks," in *Proc. IEEE 11th Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2018, pp. 162–169.
[94] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, 2014.
[95] C. Xu, J. Shen, X. Du, and F. Zhang, "An intrusion detection system using a deep neural network with gated recurrent units," *IEEE Access*, vol. 6, pp. 48697–48707, 2018.
[96] D. Kwon, K. Natarajan, S. C. Suh, H. Kim, and J. Kim, "An empirical study on network anomaly detection using convolutional neural networks," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 1595–1598.
[97] S. Baek, D. Kwon, J. Kim, S. C. Suh, H. Kim, and I. Kim, "Unsupervised labeling for supervised anomaly detection in enterprise and cloud networks," in *Proc. IEEE 4th Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, Jun. 2017, pp. 205–210.
[98] P. Mishra, K. Khurana, S. Gupta, and M. K. Sharma, "VMAnalyzer: Malware semantic analysis using integrated CNN and bi-directional LSTM for detecting VM-level attacks in cloud," in *Proc. 12th Int. Conf. Contemp. Comput. (IC3)*, Aug. 2019, pp. 1–6.
[99] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, and D. Gan, "Cloud-based cyber-physical intrusion detection for vehicles using deep learning," *IEEE Access*, vol. 6, pp. 3491–3508, 2018.
[100] A. Abusitta, M. Bellaiche, M. Dagenais, and T. Halabi, "A deep learning approach for proactive multi-cloud cooperative intrusion detection system," *Future Gener. Comput. Syst.*, vol. 98, pp. 308–318, Sep. 2019.
[101] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya, and R. Ranjan, "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 924–935, Sep. 2019.
[102] K. Wust and A. Gervais, "Do you need a blockchain?" in *Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT)*, Jun. 2018, pp. 45–54.
[103] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117134–117151, 2019.
[104] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep. 1, 2019.
[105] I. Giechaskiel, C. Cremers, and K. B. Rasmussen, "On bitcoin security in the presence of broken cryptographic primitives," in *Proc. Eur. Symp. Res. Comput. Secur.* Cham, Switzerland: Springer, 2016, pp. 201–222.
[106] Y. Li, T. Marier-Bienvenue, A. Perron-Brault, X. Wang, and G. Paré, "Blockchain technology in business organizations: A scoping review," in *Proc. 51st Hawaii Int. Conf. Syst. Sci.*, 2018, pp. 4474–4483.
[107] S. K. Lo, X. Xu, Y. K. Chiam, and Q. Lu, "Evaluating suitability of applying blockchain," in *Proc. 22nd Int. Conf. Eng. Complex Comput. Syst. (ICECCS)*, Nov. 2017, pp. 158–161.
[108] K. Zīle and R. Strazdiņa, "Blockchain use cases and their feasibility," *Appl. Comput. Syst.*, vol. 23, no. 1, pp. 12–20, May 2018.

[109] J. R. Douceur, "The Sybil attack," in *Proc. Int. Workshop Peer-Peer Syst.* Berlin, Germany: Springer, 2002, pp. 251–260.
[110] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "A survey on the scalability of blockchain systems," *IEEE Netw.*, vol. 33, no. 5, pp. 166–173, Sep. 2019.
[111] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," in *Proc. Italian Conf. Cyber Secur.*, 2018, pp. 1–11.
[112] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and A. Mohaisen, "Exploring the attack surface of blockchain: A systematic overview," 2019, *arXiv:1904.03487*. [Online]. Available: http://arxiv.org/abs/1904.03487
[113] W. Xiong and L. Xiong, "Smart contract based data trading mode using blockchain and machine learning," *IEEE Access*, vol. 7, pp. 102331–102344, 2019.
[114] N. Szabo, "Smart contracts: Building blocks for digital markets," *EXTROPY J. Transhumanist Thought*, vol. 18, no. 16, p. 2, 1996.
[115] K. Weiss and J. Schütte, "Annotary: A concolic execution system for developing secure smart contracts," in *Proc. Eur. Symp. Res. Comput. Secur.* Cham, Switzerland: Springer, 2019, pp. 747–766.
[116] Y. Huang, Y. Bian, R. Li, J. L. Zhao, and P. Shi, "Smart contract security: A software lifecycle perspective," *IEEE Access*, vol. 7, pp. 150184–150202, 2019.
[117] M. Pisa and M. Juden, "Blockchain and economic development: Hype vs. reality," *Center Global Develop. Policy Paper*, vol. 107, p. 150, Jul. 2017.
[118] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 839–858.
[119] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2013, pp. 6–24.
[120] P. Koshy, D. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using P2P network traffic," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2014, pp. 469–485.
[121] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: Characterizing payments among men with no names," in *Proc. Conf. Internet Meas. Conf. (IMC)*, 2013, pp. 127–140.
[122] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in bitcoin P2P network," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2014, pp. 15–29.
[123] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *Proc. IEEE Symp. Secur. Privacy*, May 2014, pp. 459–474.
[124] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, Jun. 2017, pp. 557–564.
[125] K. Gai, K.-K.-R. Choo, and L. Zhu, "Blockchain-enabled reengineering of cloud datacenters," *IEEE Cloud Comput.*, vol. 5, no. 6, pp. 21–25, Nov. 2018.
[126] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," 2019, *arXiv:1903.07602*. [Online]. Available: http://arxiv.org/abs/1903.07602
[127] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 375–392.
[128] M. Saad, M. T. Thai, and A. Mohaisen, "POSTER: Deterring DDoS attacks on blockchain-based cryptocurrencies through mempool optimization," in *Proc. Asia Conf. Comput. Commun. Secur. (ASIACCS)*, 2018, pp. 809–811.
[129] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2084–2123, 3rd Quart., 2016.
[130] G. Bissias, A. P. Ozisik, B. N. Levine, and M. Liberatore, "Sybil-resistant mixing for bitcoin," in *Proc. 13th Workshop Privacy Electron. Soc.*, 2014, pp. 149–158.
[131] D. Mills, J. Martin, J. Burbank, and W. Kasch, *Network Time Protocol Version 4: Protocol and Algorithms Specification*, document RFC 5925, 2010.
[132] M. Bastiaan. (2015). *Preventing the 51%-Attack: A Stochastic Analysis of Two Phase Proof of Work in Bitcoin*. [Online]. Available: http://referaat.cs.utwente.nl/conference/22/paper/7473/preventingthe-51-attack-a-stochasticanalysis-oft%wo-phase-proof-of-work-in-bitcoin.pdf
[133] S. Bag, S. Ruj, and K. Sakurai, "Bitcoin block withholding attack: Analysis and mitigation," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1967–1978, Aug. 2017.
[134] T. Bamert, C. Decker, R. Wattenhofer, and S. Welten, "Bluewallet: The secure bitcoin wallet," in *Proc. Int. Workshop Secur. Trust Manage.* Cham, Switzerland: Springer, 2014, pp. 65–80.
[135] S. Goldfeder, J. Bonneau, J. Kroll, and E. Felten, *Securing Bitcoin Wallets Via Threshold Signatures*. Princeton, NJ, USA: Princeton Univ., 2014.
[136] S. Solat and M. Potop-Butucaru, "ZeroBlock: Timestamp-free prevention of block-withholding attack in bitcoin," 2016, *arXiv:1605.02435*. [Online]. Available: http://arxiv.org/abs/1605.02435
[137] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 254–269.
[138] C. F. Torres, J. Schütte, and R. State, "Osiris: Hunting for integer bugs in Ethereum smart contracts," in *Proc. 34th Annu. Comput. Secur. Appl. Conf.*, Dec. 2018, pp. 664–676.
[139] H. Hasanova, U.-J. Baek, M.-G. Shin, K. Cho, and M.-S. Kim, "A survey on blockchain cybersecurity vulnerabilities and possible countermeasures," *Int. J. Netw. Manage.*, vol. 29, no. 2, p. e2060, Mar. 2019.
[140] A. Wang, A. Mohaisen, and S. Chen, "An adversary-centric behavior modeling of DDoS attacks," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 1126–1136.
[141] M. Vasek, M. Thornton, and T. Moore, "Empirical analysis of denial-of-service attacks in the bitcoin ecosystem," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2014, pp. 57–71.
[142] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3416–3452, 4th Quart., 2018.
[143] M. Castro and B. Liskov, "Practical Byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, Nov. 2002.
[144] O. S. Alkadi, N. Moustafa, B. Turnbull, and K.-K.-R. Choo, "An ontological graph identification method for improving localization of IP prefix hijacking in network systems," *IEEE Trans. Inf. Forensics Security*, vol. 15, no. 1, pp. 1164–1174, Aug. 2019.
[145] C. A. Vyas and M. Lunagaria, "Security concerns and issues for bitcoin," in *Proc. Nat. Conf. Cum Workshop Bioinf. Comput. Biol. (NCWBCB)*, 2014, pp. 10–12.
[146] J. L. Zhao, S. Fan, and J. Yan, "Overview of business innovations and research opportunities in blockchain and introduction to the special issue," *Financial Innov.*, vol. 2, Dec. 2016, Art. no. 28.
[147] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2014, pp. 157–175.
[148] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Commun. ACM*, vol. 61, no. 7, pp. 95–102, 2018.
[149] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 254–269.
[150] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 1–11, Jan. 2011.
[151] G. Nascimento and M. Correia, "Anomaly-based intrusion detection in software as a service," in *Proc. IEEE/IFIP 41st Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2011, pp. 19–24.
[152] U. Tupakula, V. Varadharajan, and N. Akku, "Intrusion detection techniques for infrastructure as a service cloud," in *Proc. IEEE 9th Int. Conf. Dependable, Autonomic Secure Comput.*, Dec. 2011, pp. 744–751.
[153] X. Wang, T.-L. Huang, and X.-Y. Liu, "Research on the intrusion detection mechanism based on cloud computing," in *Proc. Int. Conf. Intell. Comput. Integr. Syst.*, 2010, pp. 125–128.
[154] K. Vieira, A. Schulter, C. B. Westphall, and C. M. Westphall, "Intrusion detection for grid and cloud computing," *IT Prof.*, vol. 12, no. 4, pp. 38–43, Jul. 2010.
[155] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," 2009, *arXiv:0901.0131*. [Online]. Available: http://arxiv.org/abs/0901.0131

[156] H. A. Kholidy and F. Baiardi, "CIDS: A framework for intrusion detection in cloud systems," in *Proc. 9th Int. Conf. Inf. Technol. New Gener.*, Apr. 2012, pp. 379–385.
[157] T. Takahashi, Y. Kadobayashi, and H. Fujiwara, "Ontological approach toward cybersecurity in cloud computing," in *Proc. 3rd Int. Conf. Secur. Inf. Netw.*, 2010, pp. 100–109.
[158] J.-H. Lee, M.-W. Park, J.-H. Eom, and T.-M. Chung, "Multi-level intrusion detection system and log management in cloud computing," in *Proc. 13th Int. Conf. Adv. Commun. Technol. (ICACT)*, 2011, pp. 552–555.
[159] A. Zarrabi and A. Zarrabi, "Internet intrusion detection system service in a cloud," *Int. J. Comput. Sci. Issues*, vol. 9, no. 5, p. 308, 2012.
[160] T. Alharkan and P. Martin, "IDSaaS: Intrusion detection system as a service in public clouds," in *Proc. 12th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCgrid)*, May 2012, pp. 686–687.
[161] P. K. Rajendran, B. Muthukumar, and G. Nagarajan, "Hybrid intrusion detection system for private cloud: A systematic approach," *Procedia Comput. Sci.*, vol. 48, pp. 325–329, May 2015.
[162] J. Nikolai and Y. Wang, "Hypervisor-based cloud intrusion detection system," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2014, pp. 989–993.
[163] Y. Mehmood, M. A. Shibli, A. Kanwal, and R. Masood, "Distributed intrusion detection system using mobile agents in cloud computing environment," in *Proc. Conf. Inf. Assurance Cyber Secur. (CIACS)*, Dec. 2015, pp. 1–8.
[164] K. M. Vieira, F. Schubert, G. A. Geronimo, R. de Souza Mendes, and C. B. Westphall, "Autonomic intrusion detection system in cloud computing with big data," in *Proc. Int. Conf. Secur. Manage. (SAM)*, 2014, p. 1.
[165] N. Alexopoulos, E. Vasilomanolakis, N. R. Ivánkó, and M. Mühlhäuser, "Towards blockchain-based collaborative intrusion detection systems," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Secur.* Cham, Switzerland: Springer, 2017, pp. 107–118.
[166] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When intrusion detection meets blockchain technology: A review," *IEEE Access*, vol. 6, pp. 10179–10188, 2018.
[167] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, "ProvChain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability," in *Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGRID)*, May 2017, pp. 468–477.
[168] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.

OSAMA ALKADI (Member, IEEE) is currently pursuing the Ph.D. degree with the School of Engineering and Information Technology (SEIT), University of New South Wales (UNSW) Canberra, ACT, Australia. His primary research interests include cyber security with a focus on network and forensics security, deep learning, and blockchain technologies.

NOUR MOUSTAFA (Senior Member, IEEE) received the bachelor's and master's degrees in computer science from the Faculty of Computer and Artificial Intelligence, Helwan University, Egypt, in 2009 and 2014, respectively, and the Ph.D. degree in cyber security from UNSW, Canberra, ACT, Australia, in 2017. He was a Postdoctoral Fellow with UNSW Canberra, from June 2017 till December 2018. He is currently a Lecturer with the University of New South Wales, Canberra, and Helwan University, Egypt. His areas of interest include cyber security, in particular, network security, intrusion detection systems, statistics, deep learning, and machine learning techniques. He is also interested in designing and developing threat detection and forensic mechanisms to the Industry 4.0 technology for identifying malicious activities from cloud computing, fog computing, the IoT, and industrial control systems over virtual machines and physical systems.

BENJAMIN TURNBULL received the Ph.D. degree from the University of South Australia. He has worked for the Australian Government Defence Science and Technology Organization, initially for the Computer Network Defence and Forensics Group and later for Automated Analytics and Decision Support. He is currently a Senior Lecturer with UNSW Canberra. His research interests include novel cyber security defence strategies, cyber simulation, and understanding the physical impact of cyber attacks. As part of this, he also investigates the nexus of cyber security and kinetic effect to understand the true impacts of cyber attack, best-practice automated analysis, and visual techniques to aid decision support. This involves research in the fields of digital forensics, cyber security, knowledge representation, and visual analytics domains.