A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks

Osama Alkadi, *Member, IEEE*, Nour Moustafa, *Senior Member, IEEE*, Benjamin Turnbull, *Member, IEEE*, and Kim-Kwang Raymond Choo, *Senior Member, IEEE*

***Abstract*—There has been significant research in incorporating both blockchain and intrusion detection to improve data privacy and detect existing and emerging cyberattacks, respectively. In these approaches, learning-based ensemble models can facilitate the identification of complex malicious events and concurrently ensure data privacy. Such models can also be used to provide additional security and privacy assurances during the live migration of virtual machines (VMs) in the cloud and to protect Internet-of-Things (IoT) networks. This would allow the secure transfer of VMs between data centers or cloud providers in real time. This article proposes a deep blockchain framework (DBF) designed to offer security-based distributed intrusion detection and privacy-based blockchain with smart contracts in IoT networks. The intrusion detection method is employed by a bidirectional long short-term memory (BiLSTM) deep learning algorithm to deal with sequential network data and is assessed using the data sets of UNSW-NB15 and BoT-IoT. The privacy-based blockchain and smart contract methods are developed using the Ethereum library to provide privacy to the distributed intrusion detection engines. The DBF framework is compared with peer privacy-preserving intrusion detection techniques, and the experimental outcomes reveal that DBF outperforms the other competing models. The framework has the potential to be used as a decision support system that can assist users and cloud providers in securely migrating their data in a timely and reliable manner.**

***Index Terms*—Blockchain, cloud systems, collaborative intrusion detection, deep learning, learning-based ensemble model, privacy preservation.**

I. Introduction

THE LACK of trust in the shared virtualized infrastructure deployed in cloud environments is a major impediment to achieving secure decentralized applications. Malicious cyberattacks, such as Distributed Denial of Service (DDoS) and ransomware, target cloud-based platforms, exploiting the availability aspects of platforms and Internet-of-Things (IoT) networks. Such attacks are increasing in complexity and sophistication, resulting in disruptive consequences that can compromise data integrity, confidentiality, and availability. The ability to detect and respond to such attacks is vital to conducting necessary mitigation and limiting any damage caused to cloud services. Intrusion detection/prevention systems (IDSs/IPSs) are commonly deployed to monitor and discover those sophisticated attacks from endpoints of cloud networks, limiting the deployment of distributed IDSs/IPSs that correlate security event alerts.

Cloud systems still face sophisticated attack scenarios that also increase with the emergence of blockchain [1]. For instance, in June 2018, several blockchain cryptocurrencies (e.g., Bitcoin Gold and MonaCoin) fell victim to 51% attacks, leading to a loss of about 18 million worth of tokens [2]. The attackers exploited each cryptocurrency network and gain more than half of the total global mining hash rate. This vulnerability permitted the attackers to double-spend transactions, which consequently affected the integrity of the entire network. Moreover, in April 2016, an unknown attacker managed to drain more than 3.6 million ether, the Ethereum currency, equating to $50 million from a decentralized autonomous company using blockchain and smart contracts [3]. In August 2016, another 120 000 units were stolen from the exchange platform Bitfinex in Hong Kong worth more than $70 million [4]. Furthermore, Bitfinex also suffered a temporary shutdown of service due to several security breaches using DDoS attacks [5].

Blockchain has been utilized in various domains to facilitate trust and data privacy, in the sense of allowing participants to exchange transactions and share information while maintaining a degree of trust, integrity, and enhanced transparency. In other words, blockchain has applications beyond financial services and digital currency. The history of data sharing is stored on immutable audit trails, which can only be accessed by enterprises or privately hosted by cloud providers, with specific permissions and trust criteria.

IDS and blockchain can collectively be used on both cloud and IoT networks to discover cyberattacks and safeguard private data. Cloud-based IDSs can be categorized

Manuscript received March 25, 2020; revised May 18, 2020; accepted May 19, 2020. Date of publication May 22, 2020; date of current version June 7, 2021. The work of Kim-Kwang Raymond Choo was supported in part by the National Science Foundation under Award 1925723. *(Corresponding author: Kim-Kwang Raymond Choo.)*

Osama Alkadi, Nour Moustafa, and Benjamin Turnbull are with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia (e-mail: o.alkadi@student.unsw.edu.au; nour.moustafa@unsw.edu.au; benjamin.turnbull@unsw.edu.au).

Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security and the Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: raymond.choo@fulbrightmail.org).

Digital Object Identifier 10.1109/JIOT.2020.2996590

based on deployment locations into host- and network-based systems [7]. A host-based IDS (HIDS) runs on a host system or migration of virtual machine (VM) to monitor and inspect audit data of operating systems, such as memory and process audits. 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) system is placed at the infrastructure layer of cloud networks to monitor network traffic of all connected systems within a subnet. It can identify direct and indirect flooding, backdoor, port-scanning attacks, and suspicious malware activities [8].

Collaborative IDSs (CIDSs) are considered scalable and cost effective to inspect various cloud nodes. One of the primary concerns in the cloud is the ability to maintain data protection and trust management between multicloud service providers [9]. The cloud system of being public, distributed, and decentralized potentially leads to the challenge 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. It is quite difficult to measure the level of reputation among untrusted participants. Another major challenge is insider attacks, where malicious nodes collaborate to give false information and degrade the efficiency of alarm aggregation [9].

In addition to the discovery of attack events using CIDSs in the cloud, privacy-preserving techniques are widely used to transform, alter, or conceal original data for protecting them from unauthorized access [11]. The blockchain and smart contract technologies are common types of privacy preservation that offer authentication and integrity to cloud elements. Blockchain facilitates security trust and accountability through cryptography and consensus mechanisms. Bitcoin is considered one of the first successful implementations of a distributed cryptocurrency, where all transactions are processed without relying on third parties or agencies. This serves to safeguard data integrity and authenticity. The system architecture of cryptocurrencies is protected by extensive peer-reviewed cryptographic hash algorithms [10].

Ethereum is a cryptocurrency and a decentralized computing system, that permits developers to develop autonomous agents on a blockchain network to act as smart contracts. However, a reliable model of trust for digital smart contracts implemented in a systematic approach in blockchains is currently lacking [6]. The smart contract should give users the flexibility and transparency to view the location of their migrated data within the blockchain ledger and the ability to track audit files between clouds. Recent reports have highlighted several security flaws and attacks in blockchain and its associated technology, such as bitcoins and Ethereum [6], [23]. Developing a CIDS-based blockchain system in the cloud is essential to identify cyberattacks and achieve data privacy of IDS engines deployed at various cloud nodes, as suggested in this article.

The main contributions of this article are as follows.

1) A collaborative intrusion detection system is proposed based on a deep blockchain framework (DBF) that achieves data security and privacy in cloud networks.
2) A privacy-preserving method is suggested based on blockchain and smart contracts that are used to enable immutable data exchange and migration between multicloud services and accomplish consensus and data protection to cloud elements.
3) An intrusion detection method is proposed using a bidirectional long short-term memory (BiLSTM) deep learning algorithm to discover cyberattacks from network data migration in cloud systems.
4) The proposed system and its methods are assessed using the network data sets of UNSW-BN15 and BoT-IoT, and the system's performance is compared with several intrusion detection techniques to determine its effectiveness while deploying it to cloud.

The remainder of this article is organized as follows. Section II provides background and related studies. Section III discusses the proposed DBF. Section IV explains the CIDS using deep learning. The experimental results and discussions are described in Section V. Finally, we conclude this article in Section VI.

II. Background and Related Work

This section explains the concepts of blockchain and smart contracts for preserving cloud data, and intrusion detection utilized in the cloud. Prior studies used blockchain and IDS in the cloud are also discussed.

A. Privacy Preservation-Based Blockchain

Privacy preservation refers to the process of guarding sensitive data transactions against being accessed by unauthorized malicious parties [11]. Privacy-preserving approaches can be divided into four categories: 1) encryption based [12]; 2) differential privacy [13]; 3) perturbation based [13], [14]; and 4) blockchain-based mechanisms [15]. Blockchain is used as a public, transparent, and distributed ledger of transaction records, which would be used to secure data delivery between cloud systems. The underlying core principle of blockchain is the implementation of timestamped series of permanently linked blocks using cryptographic secure hash functions. Verified data are stored in a distributed ledger as a chain of blocks based on its timestamp [16].

Each participant in blockchain networks can observe data blocks to verify or reject them using the underpinning consensus model. When a data block is accepted and verified, the block is inserted into the chains based on its timestamp [16]. Blockchain generally uses 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 $\{0, 1\} * \longrightarrow \{0, 1\}^n$. 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. It is observed that the timestamp typically offers a discrete-time value that gradually increases while extending the chain [42].

Blockchain technology can be a replacement for trusted third-party intermediaries by using consensus methods to validate data transactions between participant entities [15]. The goal of consensus mechanisms is to ensure that transactional records are verified in networks such as those of cloud systems. Three popular consensus methods have been widely employed in blockchain applications: 1) Proof of Work (PoW); 2) Proof of Stake (PoS); and 3) practical Byzantine fault tolerance (PBFT). The PoW method works by picking a node in a distributed network to verify and adds a new block in each round of consensus by consuming a predefined amount of computational resources [16]. The main objective of computing the resources is to avoid Sybil attacks that generate a large number of fake identities performing on behalf of one entity [17]. The main challenge of the PoW methods occurs when a computing system controls more than half of the total computational resources in a network, which is defined as a 51% attack [18]. Alternatively, the PoS consensus method does not require computational power, but instead, the creator of a new block in each round is selected based on their stake in a deterministic way. The PoS method has been used in the cryptocurrency applications BlackCoin and Peercoin [19]. Third, PBFT is a popular consensus protocol currently being implemented in Hyperledger Fabric [20]. The third consensus method, PBFT, is composed of three phases: 1) preprepare; 2) prepare; and 3) commit. The PBFT method reduces the operational complexity, energy efficient, and highly practical in distributed systems, and currently implemented in Hyperledger Fabric [21].

B. Privacy Preservation-Based Smart Contracts

Smart contacts are computer programs running on top of the blockchain consensus to facilitate, verify, and negotiate contract terms between network participants. They are self-enforcing, self-verifying, and tamper-resistant transactions that are permanently sorted in a distributed ledger (i.e., it uses a blockchain as a database), where participants involved are unable to modify its programmable code. The most popular high-level language for implementing smart contracts in the Ethereum platform is called solidity. A contract in solidity is a collection of persistent data in state variables and code functions that resides at a specific location within a blockchain network. Such programs are compiled to Ethereum virtual machine (EVM) bytecode and can communicate messages between themselves as well as having the capability of performing a Turing-complete computation [39].

Smart contract security is addressed at various levels; with static code analysis, formal verification, and manual validations [**?**]. First, static code analysis, such as MythX and Slither, is used to automatically scan smart contracts from vulnerabilities, poor practices, and exploitable code patterns. Second, formal verification is used to transform the code of smart contracts into mathematical models that can be computationally proven to work as intended. This approach can be implemented at either the source code or the compiled byte-code level. Finally, utilizing manual validation tools, such as Apache Zepplin, enable interactive data analysis to perform peer code reviews. Dealing with complex problems at an early stage requires a robust multilevel technique to ensure smart contracts remain secure and function properly.

C. Intrusion Detection Systems in Cloud

An IDS aims to provide a defensive layer against malicious activities that attempt to compromise cloud computing systems. Insider and outsider attacks are still a major threat to distributed cloud environments and IDSs are often implemented as the first essential security defence mechanism [7]. IDS can be either a hardware device or a software application that monitors and inspects network systems for potential threat and policy violations [8]. Current cyber threats in the cloud are becoming highly sophisticated and detecting such threats is both costly and time consuming. Therefore, designing an effective IDS is crucial in securing IT cloud operations. CIDSs consist of multiple IDSs deployed on large distributed networks or individual hosts that communicate with each other to detect coordinated cyber attacks.

The main purpose of CIDS is to enhance the overall detection accuracy of a single IDS node by correlating attack evidence over various subnetworks. Hence, enforcing cooperation among different nodes would improve the capabilities to monitor sophisticated intrusions, such as DoS, DDoS, and malicious insiders. Furthermore, an isolated IDS would be easily bypassed by zero-day exploits or polymorphic code. Cloud-based IDSs are designed to move processing workloads into the cloud, both increasing visibility and also utilizing the elastic nature of the cloud. This better protects data privacy, infrastructure, and reputations.

Deep-learning-based techniques have also been employed in various systems to improve the performance of IDSs. These techniques allow systems to automatically learn feature representations needed for detecting attack events from large-scale data. Due to the increased demand for computational resources, recurrent and convolutional neural network algorithms gain increased attention and are recently applied in a supervised or unsupervised learning model for detecting anomalous events [40]. This is mainly attributed to their ability to find patterns from sequence data of cloud networks. While deep neural networks provide effective predicting methods relative to more traditional models (e.g., linear regression), the main disadvantage is their black-box approach. A decision from black-box models is difficult to interpret and it is challenging to identify which descriptors will be employed to interpret data patterns transparently. Other drawbacks include difficulties with the use of pretrained models on data sets that are limited to automatically optimize hyperparameters of models [41].

A recurrent neural network (RNN) based on a long short-term memory (LSTM) algorithm can also be applied to a supervised or unsupervised learning model for detecting anomalous events. Such an approach can be used to identify patterns from sequenced data of cloud and IoT networks [36]. The central concept behind RNN is that data is linked via a layer-by-layer feedback loop in long sections. There is a guided loop between its layers that improves its consistency,

enabling an internal memory to be created to log data from the previous input [34]. The RNN-based LSTM algorithm is employed in this article for discovering insider threats from cloud networks due to its capability of discovering small variations between normal and attack observations.

D. Related Work

Blockchain solutions have been used in several studies to improve trust among CIDSs in networks and cloud systems. For instance, Alexopoulos *et al.* [22] surveyed the methods of integrating CIDSs and blockchains. 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. Meng *et al.* [6] also reviewed the significance of using blockchain and its theoretical approaches that would be employed to secure CIDSs. Liang *et al.* [23] proposed a decentralized and secured data provenance framework that offers tamper-proof data blocks. This framework allows data accountability and improves data privacy and prevents inference attacks from exploiting cloud systems. Wan *et al.* [35] proposed a hybrid consensus algorithm called Goshawk, which combines multiple layers of the chain structure with many levels of PoW mining strategy and a ticket voting mechanism. This article presented that Goshawk is one of the early blockchain protocols with such properties having high efficiency and robustness against 51% attacks.

Liu *et al.* [36] proposed a framework for IoT applications for securely sharing data collections. They suggested combining Ethereum blockchain with deep reinforcement learning. Three main elements, environments, behavior, and incentives were used by the learning model. This increased the fairness ratio by more than 35% for the IoT software applications. In summary, the integration of blockchain and CIDS solutions would considerably improve security levels when they are deployed in cloud systems. Although IDS and privacy preservation have been widely used in the cloud. Integrating both systems could improve data security and privacy. In this article, a DBF is proposed to detect cyberattacks using IDS-based on a BiLSTM model that can learn at any point in time from the surrounding context and can further protect private data using privacy preservation-based blockchain and smart contract. Lopez *et al.* [38] suggested a novel architecture on how cloud services can be leveraged to predict electricity usage utilizing time-series models. The key aim of the prediction is to deal with demand purposes by distributing energy in real time. Load balancing algorithms were also proposed for managing the load in cloud data centers.

III. Proposed Deep Blockchain Framework

A. Overall Systematic Architecture

A DBF is proposed to detect cyberattacks and protect data in the cloud. The systematic architecture of the proposed framework includes four main components: 1) cloud vendor; 2) privacy-preservation-based blockchain and smart contract; 3) central coordinator unit (CCU); and 4) CIDS, as discussed below and illustrated in Fig. 1. Different types of cloud vendors and data centers are represented. These are denoted as data centers $A, B, C, \ldots, N$. They are identified as entities within a cloud network located in the blockchain network. These entities are expected to have enough cloud services to provide them with customer entities. The privacy-preservation-based blockchain and smart contract layer are different from a traditional cloud network as it incorporates a consortium blockchain; specifically, a distributed digital ledger containing the entire cloud transactions. This entity is replicated and stored in different nodes of the multicloud network, including within the CCU, data centers, or individual hosts.

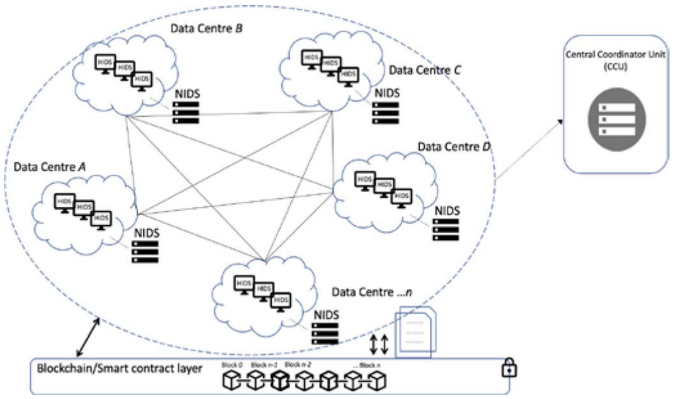

Fig. 1. Proposed cloud-based system architecture that includes the DBF. DBF would be deployed at NIDS and HIDS for cloud data centers. The blockchain/smart contract layer is designed to offer authentication and integrity to data and alerts generated by NIDS and HIDS.

The proposed DBF has been constructed in a similar data structure to the Bitcoin's structure. Mining new blocks must be sufficiently rewarded during the process of adding a block to the blockchain. The CCU acts as a SIEM tool to store IDS audit logs and alerts. By leveraging the capabilities of the CCU, incoming data from different sources (i.e., cloud data centers) are analyzed, filtered, and correlated to distinguish between normal and abnormal events. This would enable network administrators to swiftly mitigate threats and increase security awareness for participants within the blackchin cloud network. The CIDS entity orchestrates the verification of frames running on the cloud transaction network and further ensures that they adhere to the specified rules. They consist of multiple IDSs deployed on large distributed networks or individual hosts that communicate with each other to detect coordinated cyberattacks and to prevent possible illegal actions. The CIDS enforces cooperation between different nodes and would improve the capabilities to monitor sophisticated intrusions, such as DoS, DDoS, and malicious insiders.

B. Privacy Preserving Using Blockchain and Smart Contract

Blockchain-based privacy preservation in the cloud extends the idea of a blockchain protocol, which operates on a peer-to-peer aspect to deliver encrypted data transactions or network nodes in a discrete way [24]. These encrypted messages form

a chain of records or blocks that are stored on each participating cloud node confirming transaction integrity, so no records can be deleted or falsified from the ledger. Blockchain also enables the development of smart contracts, which are rule-based protocols that run on top of the blockchain network to enforce the negotiation of data usage policy (i.e., who can send IDS alerts and how they can be used) between involved cloud nodes. These policy-based rules define the raw data alerts produced by each IDS node as data transactions in the blockchain network. Collaborating IDS nodes can use the blockchain consensus mechanism to guarantee the validity and privacy of the stored alerts to create permanent and tamper-resistant data usage records.

Although blockchain and smart contract technologies remove the need for intermediaries for data protection, they are still inadequate to provide data privacy as all transactions are publicly accessible particularly in public network implementations [21]. To protect the confidentiality of smart contract data from disclosure by unauthorized users, we propose a hybrid method for privacy preserving by integrating blockchain with a trusted executed environments (TEEs). The TEE can be either hardware or software implementations that safeguard the confidentiality and integrity of applications [37]. Only permitted applications can read and write within the protected area of the CPU and memory. We propose an off-chain CIDS that log alerts in the CCU, which is protected through the implementation of TEE. The CCU operates as an isolated environment running in parallel with the blockchain network as illustrated in Fig. 1 that can also withstand attacks from other higher privileged software such as the hypervisor. The proposed framework provides a confidential platform to execute smart contracts while still ensuring integration with the existing cloud-based blockchain network such as Ethereum. To guarantee the confidentiality of the code and data of a smart contract, a secure remote assentation between the IDS nodes and CCU is established before transferring the contract.

IV. Collaborative Intrusion Detection System-Based Deep Learning

The proposed CIDS is built and deployed on could computing infrastructure due to its of their heterogeneous model and virtualized technology. Different cloud vendors may exchange event logs and shared alert data on malicious software activities amongst themselves. However, if such IDS systems 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 CIDS. These desired characteristics include efficient detection of insiders and outsiders' attacks while keeping false negatives (FNs) and false positives (FPs) at a minimum. The ability to scale dynamically across different data center networks in the entire cloud. Furthermore, the framework would provide a maximum-security resistance to mitigate zero-day vulnerabilities and ensure data confidentiality, authentication, and integrity across all CIDS nodes [8].

The architecture of a cloud-based CIDS network is depicted in Fig. 1 to represent how both HIDS and NIDS collaborate to implement instruction detection analysis at VM and network levels. Different IDSs within the same cloud domain collaborate to share data or report intrusion events based on implicit trust. However, malicious nodes can provide false information and degrade the efficiency of alarm aggregation such as in the case of collusion and betrayal attacks [9]. One emerging issue in CIDS is how to preserve data privacy, and prevent attacks from altering data, logging, or transmissions during using the event of live data migration between multicloud providers.

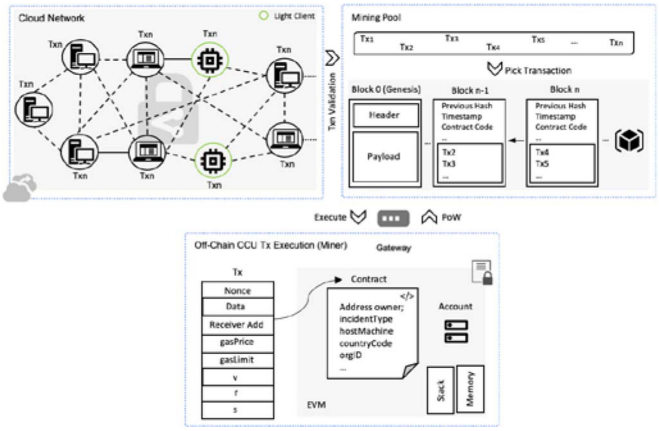

Fig. 2. Security event aggregation lifecycle of a DBF transaction.

The raw warning data generated by the IDS monitors are stored in a blockchain transaction shared between participating network nodes. This allows them to be stored for a long term in the CCU in a directory structure for further forensic analysis and compliance. Each participant (i.e., cloud data center) runs a consensus protocol to ensure the integrity of the transactions before inserting them permanently into the ledger blocks. This method ensures that the CCU database includes only valid alerts, providing tamper resistance and transparency, where cloud vendors can view the location of their migrated data within the blockchain ledger.

Participant data center nodes of the CIDS must first create a smart contract that is associated with other nodes via a registry-based type for better tracking of identification, particularly in an open public blockchain. The smart contract is sorted on each IDS within the cloud-based blackchin network. Each participating IDS node can choose which another partner node to exchange information based on defined policies within the smart contract, such as country code, region, zone, and type of organization. When a new data center or cloud vendor wishes to join the CID system, it first needs to register through the CCU to obtain an identity number (private and public keys).

The architecture of our DBF is illustrated in Fig. 2 and explained in Algorithm 1. DBF timely exchange data between CIDS, correlation of alerts and provides a full understanding of reported security events, and predictive analytics. Moreover, incorporating smart contracts in the blockchain provides a data protection mechanism to regulate identified alerts using predefined conditions. Furthermore, due to the inherent capabilities of the blockchain and smart contract-based systems, resource-constrained devices nodes are only required to store a small portion of the ledger to be processed off-chain

Algorithm 1 DBF Algorithm for Authenticating Alter Records

1. Input: blockchain transaction (t_{xn})
2. Initialisation: genesis block (index = 0 and arbitrary-length previous hash)
3. Output: newhashBlock // *return last block in chain*
4. if (t_{xn} is valid && exists) **then** // *process off-chain*
5. computeHash (hashBlock) // *create a SHA-256 hash of a block*
6. **if** (index = 0) **then**
7. hashBlock = digest (index, t_{xn}, timeStamp, previousHash, nonce)
8. **end if**
9. **return** hashBlock
10. mineBlock(last_nonce) // *CCU mining process using PoW*
11. nonce = last_nonce + 1;
12. **while** *not* (nonce (mod 99) == *n* && nonce mod(last_nonce) == 0) **do** // *difficulty of Pow, where n = number of records*
13. nonce + 1;
14. **return** nonce
15. addBlock (newblock) // *add block containing alters to chain*
16. **for** (i = 1; i < chain; i++)
17. **if** previousBlock = chain[1] **then** // *genesis block*
18. newblock = computeHash;
19. **else**
20. mineBlock(i);
21. newhashBlock = computeHash(newBlock);
22. **end if**
23. **end for**
24. **return** newhashBlock
25. end if

by the distributed CCU. Thus, IoT devices can operate as lightweight clients to verify and disseminate the correctness of alert events within a blockchain network. In contrast, the distributed CCU acts as the miner node, which stores the complete blockchain ledger, and needs to have a high hash rate to process all received transactions. Finally, the results of DBF are obtained and stored on the blockchain in sequence to ensure its immutability and reliability.

A. Deep Learning Models for Collaborative Intrusion Detection Systems

This section presents the theories and fundamentals of the proposed DBF framework. RNNs are employed as IDS for detecting attacks from a blockchain-based cloud network. It can be considered a powerful deep neural network that uses its internal memory within loops to deal with sequence data. The tackling of the temporal sequence is more relevant to the problem of intrusion detection, where the temporal patterns are present in user behavior. This would improve anomaly or outlier detection which could be difficult to infer when relying only on the spatial domain (i.e., without accounting for time dimension). For this purpose, the internal state of the RNNs is used to process sequenced lists of cryptorecords. The input sequence is handled in a series of time steps and associate memory is updated to produce a hidden state.

The standard RNN is defined as an artificial neural network with the capability of simulating discrete-time dynamical systems [34]. Such that, given a sequence of input vectors $x(t)$, a sequence of output vectors $y(t)$ is generated, where each time step $t \in [1, t_f]$ for a specific time interval t_f, one vector of the input sequence is processed, such that the internal state vectors are defined as $h_0(t) = x(t)\ \forall\ t \in [1, t_f]$ and $h_j(0) = x(t)$ $\forall j \in [1, N]$. By applying CIDS the affine transformation a_j to the output vector of the previous layer and adding the linear transformation $V_j \in R^{nj \times nj}$, the parameters of an RNN can be calculated using the cost functions

$$A_j(t) = W_j h_{j-1}(t) + V_j h_j(t-1) + b_j \tag{1}$$

$$h_j(t) = \sigma_j\big(A_j(t)\big) \tag{2}$$

where $h_j(t)$, $A_j(t) \in R^{nj}$, and $z(t) = h_j(t)$, and W_j is the weight matrix from the input layer to the hidden layer, V_j is the weight matrix between two consecutive hidden states ($h_j(t-1)$ and $h_j(t)$), b_j is the bias vector of the hidden layer, and σ_j is the activation function to generate the hidden state. The network output can be characterized as

$$y(t) = \sigma_y\big(U_j h_j(t) + b_y\big) \tag{3}$$

where U_j is the weight matrix from the hidden layer to the output layer, b_y is the bias vector of the output layer, and σ_y is the activation function of the output layer. The parameters of the RNN are trained and updated iteratively via the back-propagation method. In each time step t, the hidden layer will generate a value $y(t)$, and the last output $y(t_f)$ is the predicted network attacks.

B. Bidirectional Long Short-Term Memory Algorithm

One major drawback of RNNs is the inability of learning contextual information for an extended time caused by the vanishing gradient problem. This is mainly attributed to the prolonged temporal gap ranging from the time an input is obtained to making a decision. Weakening the ability of RNNs to learn from long-distance dependencies [25]. Therefore, the LSTM algorithm, which is an extended version of RNNs—employed the idea of gates for related units [26]. It overcomes the vanishing gradient problem, and thus allows for preserving prolonged periods of contextual information.

In this context, the view of BiLSTM originated from bidirectional RNNs [27]. Bidirectional RNNs handles sequences of the input in forward as well as backward input directions by employing two different hidden layers. Fig. 3 demonstrates a BiLSTM structure with multiple consecutive steps in time. BiLSTMs connects all hidden layers to the same output layer. A limitation of typical RNNs is that they can only use the previous context of the input data sequence. BiLSTMs compensates this by allowing for data to flow in both forward and backward directions [28].

The BiLSTM network estimates the forward hidden layer sequence output $\overrightarrow{h}(t)$, the output sequence of the backward hidden layer $\overleftarrow{h}(t)$ and the output layer $y(t)$ by reiterating the forward layer starting $t = 1$ to t_f, backward hidden layer since $t = t_f$ to 1, and then updating the final value using the following equations [29]:

$$\overrightarrow{h}(t) = H\Big(W_{\overrightarrow{j}} X_t + V_{\overrightarrow{j}} h_{\overrightarrow{j}}(t-1) + b_{\overrightarrow{j}}\Big) \tag{4}$$

$$\overleftarrow{h}(t) = H\Big(W_{\overleftarrow{j}} X_t + V_{\overleftarrow{j}} h_{\overleftarrow{j}}(t-1) + b_{\overleftarrow{j}}\Big) \tag{5}$$

$$y(t) = U_{\overrightarrow{j}} h_{\overrightarrow{j}}(t) + U_{\overleftarrow{j}} h_{\overleftarrow{j}}(t) + b_y. \tag{6}$$

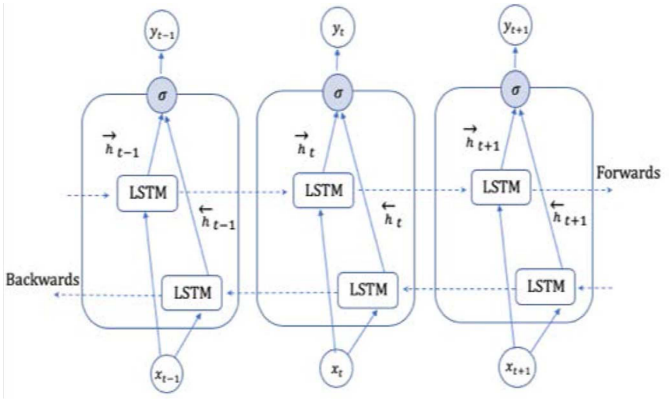

Fig. 3. Architecture of LSTM with three consecutive layers.

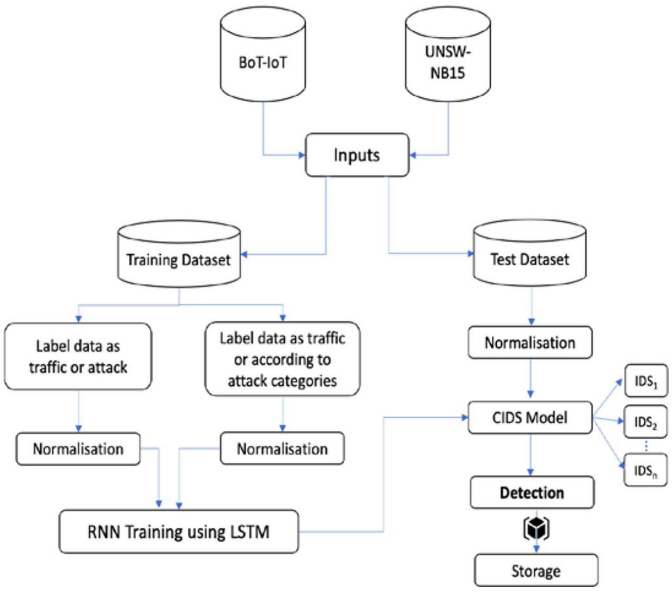

Fig. 4. Proposed collaborative intrusion detection-based BiLSTM.

The final output vector $\mathbf{y}(t)$ is calculated as

$$y(t) = \sigma_y\left(\overrightarrow{h}, \overleftarrow{h}\right). \tag{7}$$

The σ_y function concatenates the output sequences of the neurons in the hidden layers and cloud be one of four operations: 1) add; 2) multiply; 3) average; and 4) concatenate. For the RNN training stage, the BiLSTM was employed as defined by Schuster and Paliwal [27] and learning representations by backpropagating errors defined by Rumelhart *et al.* [30] (see Fig. 3).

C. Implementation of Bidirectional Long Short-Term Memory as Collaborative Intrusion Detection Systems

The details of the proposed CIDS methodology are illustrated in Fig. 4. To train and validate the BiLSTM algorithm, there are four main stages: 1) data inputs; 2) the division of training and testing sets; 3) data normalization; and 4) model development. For the first stage, data are inputted into the system. This includes data sets, for example, the UNSW-NB15 and BoT-IoT data sets. These were chosen as they involve a wide variety of security events and legitimate observations of cloud networks. In the second stage, data sets are split into training and testing sets for determining the efficiency of the BiLSTM algorithm to classify attack and normal observations. The third stage sees, the training and testing sets are normalized into a particular range, such as [0, 1], to effectively fit data using the RNN-LSTM (i.e., BiLSTM) model. We employed the min–max transformation function, due to it is the ease of scaling both data sets using the following equation:

$$x_i(j) = x_i(j) - \text{Min}(x(j))/\text{Max}(x(j)) - \text{Min}(x(j)) \tag{8}$$

where Min and Max functions represent the minimum and maximum values, respectively, from the original set for each value x_i of the feature j.

Finally, the RNN-based BiLSTM model was built to train and validate its efficiency of classifying attack events. Algorithm 2 explains the typical process of executing the RNN-based BiLSTM algorithm [29]. The BiLSTM RRN model was built using the Keras deep learning library in Python. Each data set is divided into three groups: 1) training; 2) validation; and 3) testing with a 60%; 20%; and 20% ratio, respectively. This produces the best model of accuracy detection during the implementation. The model was tested by processing each row of the testing data set by the trained model. Then, the resulting rows are classified as either normal or attack record.

Algorithm 2 RNN-Based BiLSTM Model

Input: $x(t)$
Initialization: $h_0\ (t) = x(t), \forall\ t \in [1, t_f]$
Output: $y(t)$
for $j = 1$ to N **do**
  for $t = 1$ to t_f **do**
    Save activations at each time step when performing forward pass for forward hidden layer
  end for
  for $t = t_f$ to 1 **do**
    Save activations at each time step when performing forward pass for backward hidden layer
  end for
  for $t = 1$ to t_f **do**
  end for
 end for
 for N to $j = 1$ **do**
  for t_f to $t = 1$ **do**
    For each time step save δ terms after performing backward pass for output layer only
  end for
  for $t = t_f$ to 1 **do**
    Apply stored δ terms from the output layer when performing backward pass for forward hidden layer
  end for
  for $t = 1$ to t_f **do**
    Apply stored δ terms from the output layer when performing backward pass for backward hidden layer
  end for
 end for
 return $y(t)$

TABLE I
ACCURACY OF BILSTM RNN USING THE UNSW-BN15 DATA SET WITH DIFFERENT NUMBER OF HIDDEN NODES AND HARDWARE ACCELERATORS

HN	Accuracy	Training time (s)			Testing time (s)		
		CPU	GPU	TPU	CPU	GPU	TPU
10	97.26%	200.3	59.2	54.3	13.3	8.1	7.2
20	97.61%	400.6	69.1	61.2	21.8	11.4	10.7
30	97.94%	891.2	89.5	84.7	34.3	17.4	16.9
40	98.21%	991.2	98.2	122.4	67.5	20.4	19.4
50	98.58%	1120.2	138.7	132.5	71.4	28.1	29.4
60	99.41%	1901.2	190.3	186.4	89.1	43.6	42.7

TABLE II
ACCURACY OF BILSTM RNN USING THE BOT-IOT DATA SET WITH DIFFERENT NUMBER OF HIDDEN NODES AND HARDWARE ACCELERATORS

HN	Accuracy	Training time (s)			Testing time (s)		
		CPU	GPU	TPU	CPU	GPU	TPU
10	96.71%	109.1	14.6	13.2	2.1	1.3	1.2
20	97.66%	281.4	26.1	25.3	6.4	2.9	2.6
30	97.99%	803.2	49.3	48.6	16.2	7.7	7.2
40	98.24%	981.4	79.4	64.2	31.2	15.9	15.6
50	98.51%	996.3	91.4	71.3	40.6	31.4	28.9
60	98.91%	1678.9	149.6	98.2	119.1	69.1	68.3

Central Processing Unit (CPU); Graphics Processing Unit (GPU); Tensor Processing Unit (TPU)

V. EXPERIMENTAL RESULTS AND EVALUATIONS

A. Experimental Design

To evaluate the efficiency of the proposed DBF, a private blockchain was created using Ethereum, which is an open-source blockchain platform where users can establish and deploy private blockchains within organization data sets and preprocessing module used for evaluation. The Ethereum network provides a virtual machine runtime environment to run and execute the smart contracts. The Google Colab [31] cloud service was used for the experiments using TensorFlow library Keras for deep leaning on three types of hardware accelerators CPU, GPU, and TPU for implementing the RNN-based BiLSTM model as CIDS. The CIDS model was compared with other machine learning methods; specifically support vector machine (SVM), random forest (RF), Naive Bayes (NB), and mixture localization-based outliers (MLOs) [8]. The evaluation of the proposed DBF for intrusion detection was conducted using the network data sets of UNSW-NB15 [32] and BoT-IoT [33]. The accuracy (AC), detection rate (DR), and processing time are used to evaluate the CIDS performance.

B. Results and Explanations

The RNN-based BiLSTM model as IDS was trained and validated using a large amount of labeled data in the training phase. This gives the model more information to be able to extract enough reliable features to act as a baseline for the training phase. The model was configured by an input layer fed from the two data sets, two hidden layers with hidden nodes = 60, a *tanh* activation function, and the output layer includes a *softmax* activation function to predict the two classes of normal and attack types. The model was adapted using the hyperparameters of *loss = binary_correntopy, optimizer = adam, batch_size = 100, epochs = 200, metrics = accuracy.*

The performance evaluation of the IDS was conducted on the UNSW-NB15, and BoT-IoT data sets, with the overall accuracy, demonstrated for various hidden nodes (HN = {10, 20, 30, 40, 50, 60}), as listed in Tables I and II, respectively. The training and testing computational processing times using different hardware accelerators (CPU, GPU, and TPU) are evaluated as well. The proposed framework with HN = 20 requires an approximately processing time of 400, 69, and 61 and 29, 26, and 25 s of CPU, GPU, and TPU training around 14 000 data observations for the UNSW-BN15 and BoT-IoT data sets, respectively; and 22, 11, and 11 and 6, 3, and 3 s for testing on CPU, GPU, and TPU, respectively. It is evident that the relation between estimated accuracy and number of hidden nodes is proportional; with the best accuracy (99.41% and 98.91%) achieved, respectively, with 60% hidden nodes for the UNSW-BN15 and Bot-IoT data sets.

TABLE III
DRS OF ATTACK TYPES FROM BOTH DATA SETS

Attack type	Dataset type	
	UNSW-NB15	BoT-IoT
DoS http	99.79%	99.75%
DoS udp	99.89%	99.79%
DoS tcp	99.71%	99.65%
Backdoor	97.49%	97.22%
Exploits	97.85%	96.88%
Analysis	95.65%	94.59%
Reconnaissance	96.88%	95.90%
Worms	94.75%	94.80%
Shellcode	97.90%	96.56%
Port scanning	95.98%	92.20%
OS Fingerprinting	93.14%	92.77%
DDoS http	99.89%	99.25%
DDoS udp	99.95%	99.45%
DDoS tcp	99.59%	99.10%
Keylogging	92.85%	89.90%
Data theft	98.54%	96.50%

The accuracy of the proposed model using the UNSW-NB15 and employing different hardware accelerators are shown in Tables I and II. The accuracy improves as the number of hidden nodes increases. The results revealed that with increasing hidden nodes size from 10 to 60 the accuracy of the model increases from 97.26% to 99.41% and from 96.71% to 98.91% using the UNSW-NB15 and BoT-IoT data sets, respectively. The proposed model can detect different attack types from both data sets in an average of 95%–99% as demonstrated in Table III. Various types of attacks, such as DDoS TCP, DDoS UDP, DDoS HTTP, DoS TCP, DoS UDP, and DoS HTTP, can be reliably distinguished with accuracy exceeding 99%. The proposed IDS system can also detect the remaining malicious activities that attempt to disrupt cloud services, such as reconnaissance, port scanning, keylogging, and data theft

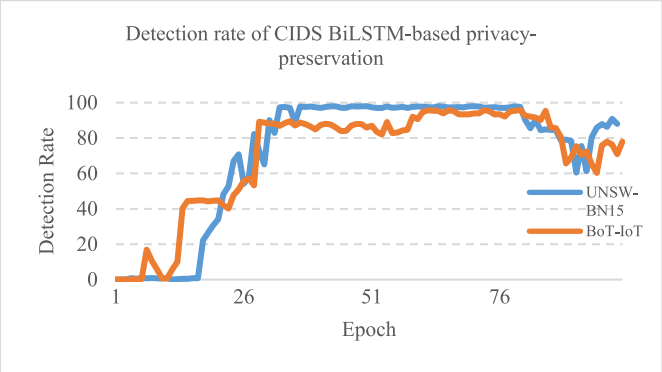

Fig. 5. Correctly classified attacks (i.e., DR) compared with the epoch times on both data sets.

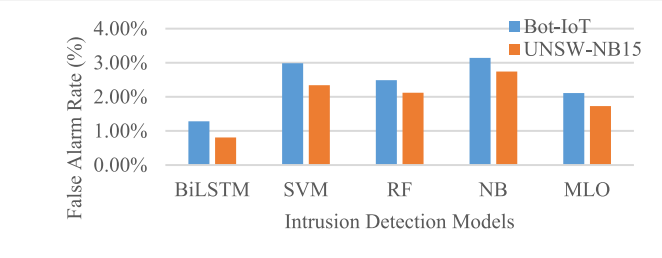

Fig. 6. Comparison of the proposed IDS-based BiLSTM model with other machine learning techniques on both data sets.

attacks, with reasonable DRs in both data sets. The model also achieves high DRs versus the epochs times in the testing phase as shown in Fig. 5.

C. Discussions

The comparison of the RNN-based BiLSTM model as CIDS with other well-known machine learning classifiers, including RF, NB, MLO, and SVM, in terms of DR under multiclass classification is shown in Fig. 6. According to the results, the false alarm rates for each attack obtained by the proposed IDS via BiLSTM records lower than the false alarm rate obtained by the RF, NB, and SVM. This shows that the model achieves the best DR compared with other models.

The RNN-based BiLSTM model can boost the performance of an IDS better than other models as it can understand the spatiotemporal context better than conventional RNNs; particularly unidirectional LSTMs. It can also prompt for quick and stable convergence while maintaining robust resilience to distortions [21]. Also, because of its adaptability and bidirectional nature, computational and memory requests can be significantly reduced. Accordingly, BiLSTMs can better differentiate the different types of attacks and further classify network abnormality by relying on its inherent ability—at any point in time—to learn from both past and future scenarios.

The proposed framework benchmarks well against other approaches for preserving privacy and identifying attack behaviors, that can be attributed to the multilevel abstraction of the data in the IDS model design. The twofold privacy model of blockchain and smart contracts can achieve perfect protection by validating data transactions and extracting features from the source data for training and validation of the IDS model. In the first phase of the privacy-based blockchain, data integrity is verified and records are checked for possible poisoning by the applied hash chain, making malicious alteration of records infeasible (i.e., highly expensive computationally). In the second phase, the data are encoded by validating unusual behavior detection as an example of efficiency and performance measuring.

The RNN-based BiLSTM technique is selected due to its inherent characterization of the spatiotemporal context, and with its synchronous and limited memory properties, it permits for accurate future conditions predications with all past and last input scenarios. Findings show that the RNN-based BiLSTM technique can efficiently classify legitimate and suspicious records after encoding the data using the two-phase privacy-preserving methods. The proposed system could be easily deployed by cloud computing centers. This could happen when the network data is called a distributed database management system to collect important features from different network nodes. It can be deployed as a service as it has a low computational overhead. This advantage stems from the fact that its potential design is based on timely estimating features' parameters of the CIDS in cloud networks.

The proposed DBF has shown its effectiveness in detecting insider and outsider attacks in both cloud and IoT environments. However, the hybrid of deep learning and blockchain-based approach can be susceptible to some disadvantages, such as communication complexity, which reflects the communication cost of propagating a new block to all parties in a system in each round. This would degrade the efficiency of alarm aggregation and discovering complex attack events in real time. Also, traffic overhead is another limitation, where a heavy network traffic environment can degrade the performance of a CIDS in handling all network packets without any drops. If the network traffic exceeds the maximum processing capability of an IDS, a large number of network packets needs to be discarded in real-time processing.

VI. Conclusion

We introduced a collaborative intrusion detection system-based DBF for the identification of cyberattacks. It is designed to also achieve privacy preservation in a cloud environment. Specifically, the privacy-preserving method comprises a hybrid approach by combining blockchain with a TEE to provide confidentiality of smart contracts, while simultaneously maintaining integrity and availability. Subsequently, the network data are encoded through a deep neural network model. The hybrid privacy-preserving method achieves better performance compared with recent works, as it is resilient to inference and data poisoning attacks. The second method of intrusion detection is based on a BiLSTM algorithm that was evaluated on the UNSW-NB15 and BoT-IoT data sets for classifying attack events that exploit cloud networks. The results revealed that the proposed intrusion detection method can outperform other techniques in terms of accuracy and DR. The proposed framework enables exchanging data between cloud simpler, safer, and more transparent while also significantly reducing

overheads. It will further act as a decision support tool to assist users and cloud providers securely migrate their data.

Future extension of this article will include applying the framework on different real-world data sets to evaluate its scalability and utility.

REFERENCES

[1] 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, pp. 1164–1174, Aug. 2019.

[2] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy rading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Trans. Depend. Secure Comput.*, vol. 15, no. 5, pp. 840–852, Sep./Oct. 2018.

[3] P. Fraga-Lamas and T. M. Fernández-Caramés, "A review on blockchain technologies for an advanced and cyber-resilient automotive industry," *IEEE Access*, vol. 7, pp. 17578–17598, 2019.

[4] J. Liu and Z. Liu, "A survey on security verification of blockchain smart contracts," *IEEE Access*, vol. 7, pp. 77894–77904, 2019.

[5] C. Baldwin. (Jul. 2019). *Bitcoin Worth $72 Million Stolen From Bitfinex Exchange in Hong Kong*. [Online]. Available: https://www.reuters.com/article/us-bitfinex-hacked-hongkong-idUSKCN10E0KP

[6] 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.

[7] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42–57, 2013.

[8] 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.

[9] W. Li, W. Meng, L.-F. Kwok, and H. Horace, "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.

[10] 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.

[11] J. B. Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. T. Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: Review and challenges," *IEEE Access*, vol. 7, pp. 164908–164940, 2019.

[12] M. S. Rahman, I. Khalil, A. Alabdulatif, and X. Yi, "Privacy preserving service selection using fully homomorphic encryption scheme on untrusted cloud service platform," *Knowl. Based Syst.*, vol. 180, pp. 104–115, Sep. 2019.

[13] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[14] T. A. Adesuyi and B. M. Kim, "A layer-wise perturbation based privacy preserving deep neural networks," in *Proc. IEEE Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, 2019, pp. 389–394.

[15] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato, "Mobile edge computing, blockchain and reputation-based crowdsourcing IoT federated learning: A secure, decentralized and privacy-preserving system," 2019. [Online]. Available: arXiv:1906.10893.

[16] 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.

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

[18] 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./Oct. 2019.

[19] I. Giechaskiel, C. Cremers, and K. B. Rasmussen, "On bitcoin security in the presence of broken cryptographic primitives," in *Proc. Eur. Symp. Res. Comput. Security*, 2016, pp. 201–222.

[20] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone. (2018). *PBFT vs. Proof-of-Authority: Applying the Cap Theorem to Permissioned Blockchain*. [Online]. Available: https://eprints.soton.ac.uk/415083/2/itasec18_main.pdf

[21] R. Huebsch *et al.*, "The architecture of PIER: An Internet-scale query processor," in *Proc. 2nd Biennial Conf. Innov. Data Syst. Res.*, 2005, pp. 28–43.

[22] N. Alexopoulos, E. Vasilomanolakis, N. R. Ivánkó, and M. Mühlhäuser, "Towards blockchain-based collaborative intrusion detection systems," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Security*, 2017, pp. 107–118.

[23] 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. Clust. Cloud Grid Comput.*, 2017, pp. 468–477.

[24] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Y. Dong, "Distributed blockchain-based data protection framework for modern power systems against cyber attacks," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3162–3173, May 2019.

[25] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, pp. 8624–8628.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[28] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, 2005.

[29] Z. Cui, R. Ke, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2018. [Online]. Available: arXiv:1801.02143.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[31] *Google Colaboratory*. Accessed: May 2019. [Online]. Available: https://colab.research.google.com

[32] 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.

[33] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic Botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.

[34] R. M. Yadav, "Effective analysis of malware detection in cloud computing," *Comput. Security*, vol. 83, pp. 14–21, Jun. 2019.

[35] C. Wan *et al.* "Goshawk: A novel efficient, robust and flexible blockchain protocol." in *Proc. Int. Conf. Inf. Security Cryptol.*, 2018, pp. 49–69.

[36] C. H. Liu, Q. Lin, and S. Wen, "Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3516–3526, Jun. 2019.

[37] P. Dalbehera *et al.*, "Dynamic configuration of trusted executed environment resources." U.S. Patent 9 264 410, Feb. 2016.

[38] J. Lopez, J. E. Rubio, and C. Alcaraz, "A resilient architecture for the smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3745–3753, Aug. 2018.

[39] X. Liu, K. Muhammad, J. Lloret, Y. W. Chen, and S. M. Yuan, "Elastic and cost-effective data carrier architecture for smart contract in blockchain," *Future Gener. Comput. Syst.*, vol. 100, pp. 590–599, Nov. 2019.

[40] 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)*, 2018, pp. 1595–1598.

[41] O. Loyola-González, "Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view," *IEEE Access*, vol. 7, pp. 154096–154113, 2019.

[42] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," in *Proc. Conf. Theory Appl. Cryptography*, 1990, pp. 437–455.