Permissioned Blockchain and Deep Learning for Secure and Efficient Data Sharing in Industrial Healthcare Systems

Randhir Kumar, *Member, IEEE*, Prabhat Kumar, Rakesh Tripathi, *Senior Member, IEEE*, Govind P. Gupta, *Member, IEEE*, A. K. M. Najmul Islam, and Mohammad Shorfuzzaman

***Abstract*—The industrial healthcare system has enabled the possibility of realizing advanced real-time monitoring of patients and enriched the quality of medical services through data sharing among intelligent wearable devices and sensors. However, this connectivity brings the intrinsic vulnerabilities related to security and privacy due to the need of continuous communication and monitoring over public network (insecure channel). Motivated from the aforementioned discussions, we integrate permissioned blockchain and smart contract with deep learning (DL) techniques to design a novel secure and efficient data sharing framework named PBDL. Specifically, PBDL first has a blockchain scheme to register, verify (using zero-knowledge proof), and validate the communicating entities using the smart contract-based consensus mechanism. Second, the authenticated data are used to propose a novel DL scheme that combines stacked sparse variational autoencoder (SSVAE) with self-attention-based bidirectional long short term memory (SA-BiLSTM). In this scheme, SSVAE encodes or transforms the healthcare data into new format, and SA-BiLSTM identifies and improves the attack detection process. The security analysis and experimental results using IoT-Botnet and ToN-IoT datasets confirm the superiority of the PBDL framework over existing state-of-the-art techniques.**

***Index Terms*—Blockchain, deep learning (DL), healthcare systems, Industrial Internet of Things (IIoT), intrusion detection system (IDS), privacy preservation.**

Manuscript received 10 January 2022; revised 9 March 2022; accepted 17 March 2022. Date of publication 23 March 2022; date of current version 9 September 2022. This work was supported in part by the Mathematical Research Impact Centric Support (MATRICS) project funded by the Science and Engineering Research Board (SERB), India, under Grant MTR/2019/001285 and in part by the Taif University Researchers Supporting Project TURSP-2020/79, Taif University, Taif, Saudi Arabia. Paper no. TII-22-0145. *(Corresponding authors: Prabhat Kumar; Govind P. Gupta.)*

Randhir Kumar, Prabhat Kumar, Rakesh Tripathi, and Govind P. Gupta are with the National Institute of Technology Raipur, Raipur 492010, India (e-mail: rkumar.phd2018.it@nitrr.ac.in; pkumar.phd2019.it@nitrr.ac.in; rtripathi.it@nitrr.ac.in; gpgupta.it@nitrr.ac.in).

A. K. M. Najmul Islam is with the LUT School of Engineering Science, LUT University, 53850 Lappeenranta, Finland (e-mail: najmul.islam@lut.fi).

Mohammad Shorfuzzaman is with the Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia (e-mail: m.shorf@tu.edu.sa).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2022.3161631.

Digital Object Identifier 10.1109/TII.2022.3161631

I. Introduction

IN RECENT years, there has been remarkable growth and development in the Internet of Things (IoT)-driven applications and services, including transportation, smart grid industry, networking, smart cities, and healthcare [1]. The extension of IoT in industrial settings, referred as Industrial IoT (IIoT) has been introduced to substantially improve the quality of conventional industries by removing geographic barriers, and enabling autonomous manufacturing, remote monitoring, and real-time data delivery to customers [2]. The current healthcare systems also take the advantage from IIoT where industrial sensors and actuators are used as wearable devices to collect users physiological data, such as blood pressure, electrocardiogram, temperature, and so on [3]. In such scenario, the data generated from industrial healthcare systems are often delivered or transmitted to patients' local gateway or edge devices to perform data processing and aggregation, and then forwarded to cloud for long-term storage, and further also used by healthcare providers for real-time diagnosis and analysis [4]. However, in the present healthcare ecosystem, the devices and sensors continuously monitor, communicate, and exchange information over insecure public channel [5]. In addition, round-the-clock connectivity of devices also makes the entire healthcare systems vulnerable to various security issues, including data manipulation, denial-of-service, eavesdropping, impersonation, man-in-the-middle, and replay attacks [6]. This raises severe concerns in the healthcare industry, as data manipulation can lead to incorrect diagnoses, potentially putting patients under observation in life-threatening scenarios [7].

Apart from this, privacy and integrity of data are other major challenges in the present industrial healthcare systems. We believe data privacy is mostly related to active data privacy attacks (ADPAs) and passive data privacy attacks (PDPAs) [8]. In ADPA attack, the attacker tries to alter/modify or infer private data during data transfer between two communicating entities (such as data poisoning attacks) [9]. These attacks are launched to modify real-time patient's health data. Moreover, it can negatively impact the performance of the artificial intelligence-based data analytic or attack detection process of intrusion detection systems (IDSs) [10]. On the other hand, PDPA is launched by the attacker to sniff (private) data, i.e., to gain some fundamental statistical properties from the training dataset (such as data

1551-3203 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on February 12,2023 at 01:59:57 UTC from IEEE Xplore. Restrictions apply.

inference attacks) [11]. Moreover, privacy breaches are also related to authentication, i.e., a condition where unauthenticated medical sensors can easily be used as a surveillance device to track and/or monitor critical information of patients unknowingly [12]. As a result, an efficient authentication scheme for controlling participating IoT devices is also required, which can be used to minimize authentication-related privacy breaches [13].

A. Threat Model

The widely adopted the "Dolev–Yao (DY) threat model" [14] is used in designing the proposed PBDL framework. According to the "DY model" the communicating entities (i.e., IoT devices, edge nodes, and cloud vendors) are not fully trustworthy and data sharing is done over insecure public channels. As a result, the data exchanged between the communicating entities can be intercepted, modified (i.e., data poisoning attack), deleted, or even malicious contents can be injected during communication. Apart from the "DY model," the current *de facto* "Canetti and Krawczyk's adversary model," [15] known as the "CKadversary model," is also utilized in designing the PBDL framework. According to the "CKadversary model," an attacker "$\mathcal{A}$" can gain access to the secret credentials as well as the "session keys (session states)" for a certain session. Similar to the "DY model," in the "CKadversary model" edge and cloud nodes are considered as semitrusted entities and registration authority is assumed as the trusted entity in the network [8].

B. Key Contribution

In this article, we design and implement a permissioned blockchain and deep learning (DL) techniques for enabling secure and efficient data sharing in industrial healthcare systems. The following are the major contributions of this article.

1) Permissioned blockchain and smart contracts are combined with DL techniques to design a novel framework called PBDL. The underlying framework provides a secure and efficient mechanism to transmit healthcare data between device–edge–cloud.
2) In PBDL, a blockchain scheme is designed that first registers the participating entities, then verifies them using the zero-knowledge proof (ZKP) identification system, and finally validates using the smart contract-based consensus mechanism. The underlying approach enables immutable data exchange and prevents data from poisoning attacks. An interplanetary file system (IPFS)-based off-chain storage is also integrated to achieve high throughput and scalability during real-time data access.
3) The authenticated data are used by the proposed DL scheme. The latter combines stacked sparse variational autoencoder (SSVAE) with the self-attention-based bidirectional long short term memory (SA-BiLSTM) model to form a new DL architecture. In this scheme, SSVAE is employed to transform actual industrial healthcare data into new format in an unsupervised manner (i.e., to prevent inference attack). The encoded data are further used by the SA-BiLSTM technique for intrusion detection. We employed an attention mechanism to concentrate more on the information extracted from the forward and backward hidden layers of BiLSTM.

The rest of this article is organized as follows. Section II provides related work. Section III presents the proposed framework. The security analysis is performed in Section IV. The experimental results are presented in Section V. Finally, Section VI concludes this article.

II. Related Work

In order to overcome the aforementioned challenges, various solution related to blockchain have been proposed in the literature [16], [17]. For example, Tandon *et al.* [18] highlighted the importance of security and privacy in the healthcare system, and suggested the advantage and challenges of utilizing blockchain as a solution in the healthcare system. Farouk *et al.* [6] illustrated the need of data privacy protection in the IoT-enabled healthcare system, and emphasized on how the blockchain technology is used to achieve privacy goals. Al-Turjman *et al.* [4] discussed different ways to integrate blockchain with the healthcare system in order to address issues, such as security, privacy, access control integrity, and ownership. Gupta *et al.* [19] reviewed the benefits of smart contracts in terms of privacy protection and how they can extend the capabilities of blockchain. Some researches [20], [21] are targeted at illustrating the benefits of blockchain-based smart healthcare systems and recommended various security designs but lack implementation specific details. Xu *et al.* [2] proposed the privacy-protection model for healthcare data based on fine-grained access control and blockchain. Rahman *et al.* [22] presented a secure and provenance enhanced framework for healthcare systems based on federated learning and differential privacy. In this approach, blockchain and smart contracts perform the trust management, edge training, and authenticates the federated participating entities.

Several studies have been proposed and used to preserve privacy of data along with the application of intrusion detection in IoT and industrial healthcare systems [1], [9], [11], [23], [24]. Various researchers used machine learning (ML)- and DL-based techniques to design IDS in healthcare systems. For instance, Kumar *et al.* [7] designed an IDS based on the fog-cloud architecture and ensemble learning in healthcare environment. Begli *et al.* [25] proposed a secure IDS (SVM-IDS) in remote healthcare systems. Furthermore, Priya *et al.* [26] presented a deep neural network-based IDS for the healthcare system. In this model, features were extracted using grey wolf optimization and principal component analysis. Newaz *et al.* [27] proposed HealthGuard that applied different ML approaches to provide security and privacy. He *et al.* [28] developed a DL-based IDS that secured healthcare systems using the stacked autoencoder approach. All these solutions have proved that DL-based IDS can achieve better performance compared to ML-based IDS.

III. Proposed Framework

A. Overall Systematic Architecture

The systematic architecture of the proposed PBDL framework is made up of three layers: 1) the industrial healthcare system layer; 2) the edge-blockchain layer; and 3) the cloud-blockchain layer, as shown in Fig. 1.

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on February 12,2023 at 01:59:57 UTC from IEEE Xplore. Restrictions apply.

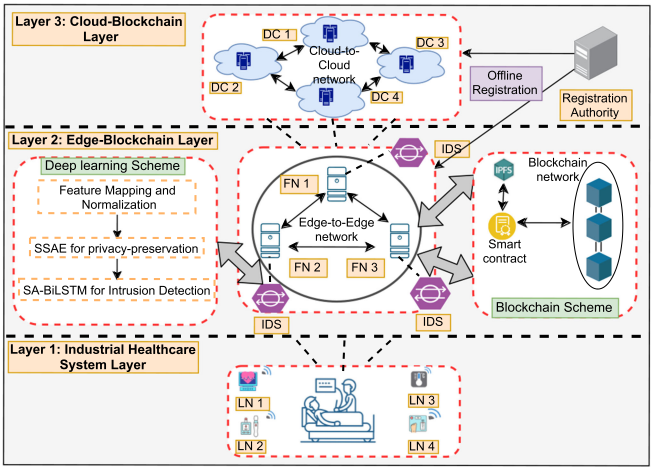

Fig. 1. System model.

1) Industrial Healthcare System Layer: In this layer, various IoT-based healthcare tracking systems and implantable medical devices (e.g., temperature sensors, glucose monitor, and heart rate devices) are used to continuously collect the patients important health information. As these devices have limited resources and computational capacity, they can only keep and process a portion of the data on the blockchain, hence they are denoted as *lightweight nodes* (LNs).

2) Edge-Blockchain Layer: This layer is made up of powerful nodes, such as data analysis servers, industrial computers, edge-computing servers, and so on, and is referred as *full node* (FN). The peer-to-peer network is built through geo-distributed regions of edge devices located in primary and urban health centres. Each patient (i.e., an associated healthcare device) is assigned with edge service node that collects, process, and raises alarms in emergency situations, and interacts with cloud for long-term storage and backup.

3) Cloud-Blockchain Layer: This layer includes a number of cloud providers or vendors and data centres (DCs). These DCs are in charge of providing clients with services (such as computation, processing, and so on).

In a typical healthcare system, we mainly have LNs, FNs, and DCs as communicating entities. In such a scenario, LNs have limited resources and can only communicate data to FNs over the edge-blockchain layer. The FNs assist LNs to search, mine, and add a transaction in the blockchain network. Finally, DCs are in charge for storing data from FNs for as long-term storage. Data are sent to the edge by the DCs as per the requirement. The proposed blockchain technique is first used to register all three participating nodes, then verified using the ZKP protocol, and finally, the smart contract-based consensus-enhanced proof-of-work mechanism, mentioned in [23], is used to authenticate data transactions in the network. Furthermore, the IPFS-distributed storage layer is utilized to keep complete transactions, and the produced hash is being kept on the blockchain network, mentioned in [12]. Finally, the DL technique is first used to convert the authenticated data, and then identify intrusions in the network. At various network nodes, this technique is delivered as software-as-a-service (i.e., edge servers and cloud DCs, gateways, and routers) mentioned in [7]. Furthermore, the framework is installed on a large-scale distributed network model or an individual host that successfully communicates with one another over the edge-blockchain and cloud-blockchain layers, and it coordinates with one another for the detection of cyberattacks.

B. Blockchain Scheme

In the proposed PBDL framework, a permissioned blockchain is designed due to following two principal reasons.

1) First, as we mentioned in Section I, privacy-preservation in terms of sharing of data within a set of known and authorized parties is a fundamental consideration.
2) Second, permissionless blockchains are open and can increase attacks from external adversaries without a notable function-enhancement. Therefore, in our opinion, these are unsuitable in industrial healthcare systems.

The proposed scheme has four different phases: 1) system registration; 2) verification using ZKP ($\mathcal{ZKP}$); 3) validation and block creation phase; and 4) data generation and block updation phase. We have discussed the steps and working of each phase as follows.

1) Registration Phase: In this step, a trusted registration authority registers the DC ($\mathcal{DC}_j$) and FN ($\mathcal{FN}_j$) safely in the offline mode. In addition, using the ZKP ($\mathcal{ZKP}$) protocol, light nodes, or sensor nodes ($\mathcal{LN}_j$), are registered with ($\mathcal{FN}_j$). This protocol verifies the identification of two people without providing any personal or confidential information. In this strategy, one side assumes the role of challenger, while the other assumes the role of prover. It becomes a verified party if the prover's response is valid. After ZKP verification, ($\mathcal{FN}_j$) registers $\mathcal{LN}_j$ by launching a request $\mathcal{R}_j$. The following are the steps in the registration and verification process.

1) *Step 1:* In the first stage, ($\mathcal{FN}_j$) creates a temporary key ($\mathcal{T}_{kj}$) for $\mathcal{LN}_j$, which is made up of three major components: 1) sensor temporary identification of $\mathcal{LN}_j$ ($\mathcal{TID}_j$); 2) MAC of sensor ($\mathcal{S}_j^{\mathcal{MAC}}$); and 3) geographical location of sensor ($\mathcal{G}_j^{\mathcal{LOC}}$). It is worth mentioning here that in the industrial system, one patient can have varying number of sensors with its types. Thus, $\mathcal{S}_j^{\mathcal{MAC}_m}$ comprises of m MACs associated with individual industrial sensors. Furthermore, the timestamp ($\mathcal{T}_{sj}$) of $\mathcal{T}_{kj}$ is recorded for the request generation of registration process. After this, the $\mathcal{T}_{kj}$ gets forwarded to ($\mathcal{FN}_j$) using a secure channel.
2) *Step 2:* Once ($\mathcal{FN}_j$) receives $\mathcal{T}_{kj}$, it extracts the included parameters. Then, $\mathcal{SALT\text{-}P}$ generates pseudorandom number, which gets appended to $\mathcal{T}_{kj}$, to protect from precomputed hash attack. To create unique hash, a SALT adds random bits before hash computation, which prevents from attack of precomputed hash. Furthermore, the $\mathcal{SALT\text{-}P}$ and $\mathcal{T}_{kj}$ merged together for hash computation by SHA-2. The SHA-2 creates 256 bits of hash, which is computationally inexpensive comparing to other hash techniques. After this, hash $\mathcal{H}[\mathcal{ID}_j]$ is passed to $\mathcal{LN}_j$ along with public key of $\mathcal{FN}_j(PK_j)$ using a secure channel.

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on February 12,2023 at 01:59:57 UTC from IEEE Xplore. Restrictions apply.

2) Verification $\mathcal{ZKP}$ Phase: The $\mathcal{ZKP}$ approach is used here for verification from the end of the light node ($\mathcal{LN}_j$). $\mathcal{LN}_j$ uses $\mathcal{SALT}$-$\mathcal{P}$ and $\mathcal{T}_{kj}$ to discover the right $\mathcal{H}[\mathcal{ID}_j]$ utilizing $\mathcal{SALT}$-$\mathcal{P}$ and $\mathcal{T}_{kj}$ combination after receiving information from ($\mathcal{FN}_j$). The light node uses new $\mathcal{SALT}$-$\mathcal{Q}$ to compute and maintains a difficulty level by appending the value of (z). Large prime value ($\mathcal{S}$) and matching generator ($\mathcal{T}$) are used to calculate the value z. In this case, ($\mathcal{FN}_j$) signifies as a prover and ($\mathcal{LN}_j$) is verifier. The $\mathcal{LN}_j$ has to provide a prove of secret timestamp [($\mathcal{T}_{\delta j}$)], as $y = T^{\mathcal{T}_{\delta j}} \bmod S$. But, value of ($\mathcal{T}_{\delta j}$) cannot be revealed in the complete process. As the complete process is highly depends on the understanding of ($\mathcal{T}_{\delta j}$) without any disclose. The $\mathcal{LN}_j$ finds random number (u), which is further used in the computation of $v = T^u \bmod S$. The d value is applied to create $\mathcal{H}_i$ by $\mathcal{SALT}$-$\mathcal{Q}$ and sent to ($\mathcal{FN}_j$). Furthermore, created $\mathcal{H}_i$ gets encrypted by $\mathcal{PK}_j$ to evaluate $\mathcal{ZKP}$ and communicated to ($\mathcal{FN}_j$). After obtaining $\mathcal{ZKP}$, the ($\mathcal{FN}_j$) decrypts by secret key and extract the value of T^u mod S such that the hash of T^u mod S and $\mathcal{SALT}$-$\mathcal{Q}$ is identical to $\mathcal{ZKP}$. Now, the ($\mathcal{FN}_j$) asks the question (g_1)= ($\mathcal{T}_{\delta j+\mathcal{R}}$) to create hash by $\mathcal{SALT}$-$\mathcal{Q}$ for transmit to $\mathcal{LN}_j$. Once the $\mathcal{LN}_j$ receives the q_1, then it computes the answer A_1, and appends the $\mathcal{SALT}$-$\mathcal{Q}$ hash and transmit it to ($\mathcal{FN}_j$) for verification. Finally, the answer is verified by the ($\mathcal{FN}_j$), if it is found correct then it assigns permanent id ($\mathcal{ID}_j$) to $\mathcal{LN}_j$. The same $\mathcal{ID}_j$ is disseminated to the blockchain network for further communication. The entire process of registration and verification is summarized in Table I.

3) Validation and Block Creation Phase: The joining of the blockchain process occurs if the $\mathcal{LN}_j$ is registered by the $\mathcal{FN}_j$. The validation and block formation processes are given in the Table II. These are the procedures involved in creating and validating $\mathcal{LN}_j$ and $\mathcal{FN}_j$ blocks.

Step 1: In the initial step, $\mathcal{LN}_j$ produces a key value pair (PB_{kj} and PR_{kj}) with PB_{kj} being the public key, whereas PR_{kj} being the private key of the jth light node ($\mathcal{LN}_j$). In addition, $\mathcal{FN}_j$ starts the registration process.

Step 2: The $\mathcal{FN}_j$ generates a signature ($\boldsymbol{g}_j$) and sends it to the appropriate $\mathcal{LN}_j$ for validation.

Step 3: The $\mathcal{LN}_j$ validates a signature. When the $\boldsymbol{g}_j$ matches successfully, $\mathcal{LN}_j$ sends a request to join the network to $\mathcal{FN}_j$ with the credentials PB_{kj}, $\mathcal{GLOC}_j$, and $\mathcal{ID}_j$.

Step 4: Then, for validation of the location of $\mathcal{LN}_j$, $\mathcal{FN}_j$ transmits $\mathcal{GLOC}_j$ to the peer nodes ($\mathcal{FN}_j$).

Step 5: The peer nodes ($\mathcal{FN}_j$) use smart contracts to validate the position of $\mathcal{LN}_j$ using the timestamp recoded by $\mathcal{FN}_j$, as well as the latitude and longitude of $\mathcal{LN}_j$. After it has been confirmed, the appropriate node receives a True/False acknowledgment.

Step 6: A latest block (B_j) is constructed and attached to the network of blockchain, including credential PB_{kj} and $\mathcal{ID}_j$, for True status.

4) Data Creation and Updation of Block: This phase includes data generation by $\mathcal{LN}_j$. The Table III gives a summarized view of data generation and updation of block. The created data are

TABLE I
REGISTRATION AND VERIFICATION OF $\mathcal{LN}_j$

Light Node ($\mathcal{LN}_j$)	Full Node ($\mathcal{FN}_j$)
INPUT: R_j OUTPUT: $\mathcal{ID}_j$ $\mathcal{T}_{kj} \rightarrow (\mathcal{T}_j^{\mathcal{ID}}, \mathcal{S}_j^{\mathcal{MAC}_m}, \mathcal{G}_j^{\mathcal{LOC}})$ Store ($\mathcal{T}_{\delta j}$) $\leftarrow$ Timestamp of $\mathcal{T}_{kj}$ $(\mathcal{T}_{kj}(R_j) = \mathcal{T}_j^{\mathcal{ID}}, \mathcal{S}_j^{\mathcal{MAC}}, \mathcal{G}_j^{\mathcal{LOC}})$ $\xrightarrow{\quad\quad}$ (SSL/TLS)	
	Extract: $\mathcal{T}_{kj}(R_j) \rightarrow \mathcal{T}_j^{\mathcal{ID}}, \mathcal{S}_j^{\mathcal{MAC}_m}, \mathcal{G}_j^{\mathcal{LOC}}$ $\mathcal{H}[\mathcal{ID}_j] = \mathcal{H}(\mathcal{T}_{kj} \mid\mid \mathcal{SALT}$-$\mathcal{P})$ distribute public key: $\mathcal{PK}_j$ $(\mathcal{H}[\mathcal{ID}_j], \mathcal{PK}_j)$ $\xleftarrow{\quad\quad}$ (SSL/TLS)
Verification of ZKP Compute: $\mathcal{SALT}$-$\mathcal{P}$ such that: $\mathcal{H}(\mathcal{T}_{kj} \mid\mid \mathcal{SALT}$-$\mathcal{P}) == \mathcal{H}[\mathcal{ID}_j]$ $\mathcal{SALT}$-$\mathcal{P}$ generates pseudo random number Evaluate y= $T^{\mathcal{T}_{\delta j}}$ *mod* S Evaluate u $\leftarrow \mathcal{T}_{\delta j} + \mathcal{R}_j$ Evaluate v=T^u *mod* S Hash: $\mathcal{H}_j = \mathcal{H}$(v $\mid\mid \mathcal{SALT}$-$\mathcal{Q}$) $\mathcal{ZKP} = \mathcal{PK}_j(\mathcal{H}_j)$ ZKP $\xrightarrow{\quad\quad}$ (SSL/TLS)	
	Decrypt: $\mathcal{SK}_j(\mathcal{ZKP})$ Retrieve $\mathcal{ZKP}$: v=T^u *mod* S Find $\mathcal{SALT}$-$\mathcal{Q}$ so that $\mathcal{H}$(v $\mid\mid \mathcal{SALT}$-$\mathcal{Q}$)==$\mathcal{ZKP}$ Ask $g1 = \mathcal{T}_{\delta j+\mathcal{R}}$ OR $g2 = (\mathcal{T}_{\delta j+\mathcal{R}} + \mathcal{T}_{\delta j}) \bmod$(S-1) such that: $(\mathcal{T}_{\delta j+\mathcal{R}}) = (\mathcal{T}_{\delta j+\mathcal{R}} + \mathcal{T}_{\delta j})$ Send $g1/g2$ $g = \mathcal{H}(g1/g2 \mid\mid \mathcal{SALT} - \mathcal{Q})$ g $\xleftarrow{\quad\quad}$ (SSL/TLS)
Extract: $g \rightarrow g1/g2$ if (g==g1) then Send $\mathcal{A}_1 \leftarrow \mathcal{T}_{\delta j+\mathcal{R}}$ Else Send $\mathcal{A}_2 \leftarrow (\mathcal{T}_{\delta j+\mathcal{R}} + \mathcal{T}_{\delta j}) \bmod$(S-1) Send $\mathcal{A}_1$ / $\mathcal{A}_2$ $\mathcal{A} = \mathcal{H}(\mathcal{A}_1/\mathcal{A}_2 \mid\mid \mathcal{SALT} - \mathcal{Q})$ A $\xrightarrow{\quad\quad}$ (SSL/TLS)	
	Extract $\mathcal{A}$ Compute A by using $\mathcal{SALT} - \mathcal{Q})$ to compute $\mathcal{A}_1/\mathcal{A}_2$ if ($\mathcal{A}$==$\mathcal{A}_1$) verify $T^{\mathcal{T}_{\delta j+\mathcal{R}}}$ *mod* S==v else Compute $T^{\mathcal{T}_{(\delta j+\mathcal{R})} Mod(\mathcal{S}-1)}$ *Mod* S== v.y mod S successful verification Assign $\mathcal{LN}_j \leftarrow \mathcal{ID}_j$ Add $\mathcal{LN}_j$ to blockchain network Disseminate the $\mathcal{ID}_j$ to other peer

TABLE II
PROCESS OF BLOCK CREATION AND VALIDATION

Light Node ($\mathcal{LN}_j$)	Full Node ($\mathcal{FN}_j$)	Full Node ($\mathcal{FN}_j$) peer
INPUT: $\mathcal{ID}_j$ OUTPUT: B_j Create key pair (PB_{kj},PR_{kj}) of $\mathcal{ID}_j$ PB˙kj $\xrightarrow{\quad\quad}$ (SSL/TLS)		
	Create δig_j sig˙j $\xleftarrow{\quad\quad}$ (SSL/TLS)	
Validate (δig_j) Send (PB_{kj},$\mathcal{G}_j^{\mathcal{LOC}}$) (Request to join PB˙kj,G˙LOC˙j, ID˙j) $\xrightarrow{\quad\quad}$ (SSL/TLS)		
	Send $\mathcal{G}_j^{\mathcal{LOC}}$ to $\mathcal{FN}_j$ peer nodes G˙LOC˙j $\xrightarrow{\quad\quad}$ (SSL/TLS)	
		Check $\mathcal{G}_j^{\mathcal{LOC}}$ with timestamp, latitude, and longitude of $\mathcal{LN}_j$ Return status (True/False) (True) $\xleftarrow{\quad\quad}$ (SSL/TLS)
	Validates $\mathcal{G}_j^{\mathcal{LOC}}$ if true, create block B_j Append the B_j to blockchain Distributes B_j to peer nodes $\mathcal{FN}_j$ (B˙j) $\xrightarrow{\quad\quad}$ (SSL/TLS)	

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on February 12,2023 at 01:59:57 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PROCESS OF DATA CREATION AND UPDATION OF BLOCK

Light Node ($\mathcal{LN}_j$)	Full Node ($\mathcal{FN}_j$)	Full Node ($\mathcal{FN}_j$) peer
INPUT: $\mathcal{D}ata_j$ OUTPUT: Update B_j read ($\mathcal{D}ata_j$) which includes $\mathcal{S}_j^{\mathcal{MAC}}$, PB_{kj}, $\mathcal{ID}_j$, and $\mathcal{G}_j^{\mathcal{LOC}}$ of $\mathcal{LN}_j$ Sign ($\mathcal{S}ig_j$)=($\mathcal{D}ata_j$, PR_{kj}) Create T_j= ($\mathcal{D}ata_j$, PB_{kj}, $\mathcal{S}ig_j$,$\mathcal{ID}_j$) Send T_j		
$\xrightarrow[\text{(SSL/TLS)}]{\text{(T`j)}}$		
	Validate $\mathcal{ID}_j$, PB_{kj} Check $\mathcal{D}ata_j$, $\mathcal{S}ig_j$ Validate $\mathcal{S}ig_j$ Add T_j to B_j Disseminate to $\mathcal{FN}_j$ peer	
	$\xrightarrow[\text{(SSL/TLS)}]{\text{(B`j)}}$	
		Validate B_j Synchronized Blockchain

identified as transactions (T_j). The entire process of data creation and alteration in a block is discussed as follows.

1) *Step 1:* At first step, data ($\mathcal{D}ata_j$) are generated by $\mathcal{S}_j^{\mathcal{MAC}}$ and signed ($\boldsymbol{g_j}$) with PR_{kj} of $\mathcal{LN}_j$, after successful signature of $\mathcal{D}ata_j$, a new transaction (T_j) gets generated including the credentials $\boldsymbol{g_j}$, PB_{kj}, and $\mathcal{ID}_j$ of $\mathcal{LN}_j$. Furthermore, T_j gets forwarded to $\mathcal{FN}_j$ for its validation and updation in B_j.
2) *Step 2:* Furthermore, PB_{kj} gets associated with $\mathcal{ID}_j$ and valid record gets verified with credentials $\mathcal{D}ata_j$ and $\boldsymbol{g_j}$. Once the required credential matches successfully, then T_j gets added into a block B_j and shared over the blockchain network. At last, B_j gets updated and appended into the blockchain network.

C. DL Scheme

Once the communicating entities are registered and validated in the network. The proposed DL scheme is enforced on the authenticated data to detect intrusions. The proposed scheme first performs feature mapping and data normalization using steps mentioned in [7] and [23]. Then, we design a SSVAE technique to reshape or encode data (used to prevent inference attacks), and the encoded data are finally used by the proposed SA-BiLSTM for intrusion detection.

The VAE technique works on the principal of the graphical model with the directed probabilistic approach, which is implemented at this stage and is achieved by approaching the neural network posterior. Let's say that we have the actual $\mathcal{F} = \{\boldsymbol{a}_j\}_{j=1}^{B}$ dataset that contains the $\boldsymbol{a}$ and N record attributes. The latent variable $\mu's$ is used by the VAE, and then characterizes the $\mathcal{F}$ distribution. We presume that the conditional distribution of the latent variable μ denotes the Gaussian distribution (GD) [8]. In addition, the theory shows that if the hidden variable μ matches GD, neural network produces data from a distribution. This means that a new dataset $\widehat{\mathcal{F}} = \{\widehat{\boldsymbol{a}_j}\}_{j=1}^{B}$ is generated by μ by optimizing the generated Ω parameter, which is pretty much the same as the original dataset $\mathcal{F} = \{\boldsymbol{a}_j\}_{j=1}^{B}$. This indicates that $q_\Omega(\boldsymbol{a})$ is a marginal probability that we want to maximize.

$$q_\Omega(\boldsymbol{a}) = \int q_\Omega(\mu) q_\Omega(\boldsymbol{a}|\mu) f\mu, \text{with } \mu \sim N(0,1). \quad (1)$$

Since the exact true posterior density of $q_\Omega(\boldsymbol{a}|\mu)$ is intractable, the VAE utilizes the $s_\Omega(\boldsymbol{a}|\mu)$ recognition model to approximate the undetermined true posterior of $q_\Omega(\boldsymbol{a}|\mu)$ to address the issue. Kullback–Leibler (KL) divergence is used in the case of VAE to calculate the relationship between the $s_\Omega(\boldsymbol{a}|\mu)$ recognition model and the actual $m_\Omega(\boldsymbol{a}|\mu)$ posterior distribution.

$$\log q_\Omega\left(\boldsymbol{a}^{(j)}\right) = D_{\text{KL}}\left(s_\Omega(\mu|\boldsymbol{a}^{(j)}) \,||\, q_\Omega(\mu|\boldsymbol{a}^{(j)})\right) + H\left(\Omega, \delta; \boldsymbol{a}^{(j)}\right). \quad (2)$$

The KL divergence must be greater than $0, \log q_\Omega(\boldsymbol{a}^{(j)}) \geq H(\Omega, \delta; \boldsymbol{a}^{(j)})$. The variational lower bound formula $G(\Omega, \delta; \boldsymbol{a}^{(j)})$ on the marginal probability of data point "j" is defined as

$$\mathcal{L}\left(\Omega, \delta; \boldsymbol{a}^{(j)}\right) = -D_{\text{KL}}\left(s_\Omega(\mu|\boldsymbol{a}^{(j)}) \,||\, q_\Omega(\mu)\right) + Y_{c\delta(\mu|\boldsymbol{a}^{(j)})}\left[\log q_\Omega\left(\boldsymbol{a}^{(j)}|\mu\right)\right]. \quad (3)$$

In order to optimize $\log q_\Omega(\boldsymbol{a})$, the marginal variational lower bound reflect the entire VAE optimization target. The first term on the right-hand side of (3) is equal to the regularization term, and the second term is a negative reconstruction error. The $q_\Omega(\boldsymbol{a}^{(j)}|\mu)$ distribution is considered to be Gaussian; therefore, it is necessary to view $s\Omega(\mu|\boldsymbol{a}^{(j)})$ as probabilistic rather than binary performance. Thus, $s_\Omega(\mu|\boldsymbol{a}^{(j)})$ is a probabilistic encoder that contains the δ variance parameter, and the $q_\Omega(\boldsymbol{a}^{(j)}|\mu)$ probabilistic decoder with the Ω generation parameter. We have extended the traditional VAE by adding $L1$ regularization, i.e., the sparse constraint in the loss function of original VAE as follows:

$$\mathcal{L}\left(\Omega, \delta; \boldsymbol{a}^{(j)}\right) = \mathcal{L}\left(\Omega, \delta; \boldsymbol{a}^{(j)}\right) + \eta_1 \mathcal{R}_1 = \mathcal{L}\left(\Omega, \delta; \boldsymbol{a}^{(j)}\right) + \eta_1 \sum_{ij} \|\boldsymbol{a}_i - \boldsymbol{a}_j\|^2 W_{ij} \quad (4)$$

where $\mathcal{R}_1$ represents the Laplacian regularization term, η_1 is the adjusting parameter of $L1$ regularization, and W is the weighting value. The SSVAE is a neural network made out of numerous layers of simple SVAE, each with its outputs connected to the inputs of the next layer.

Furthermore, to verify the effectiveness of the SSVAE-based privacy-preservation approach, a utility system based on SA-BiLSTM is designed to retain data privacy. The traditional BiLSTM initially originated from the recurrent neural network (RNN) architecture. By using two distinct hidden layers, bidirectional RNNs handle input sequences in both of the input direction, i.e., forward and backward. Typical RNNs are limited by the fact of using only the previous context of the input datasets. BiLSTMs offer by allowing the data flow in both directions (forward and backward) [9]. The BiLSTM network calculates the output of forward pass (from past to future) $\vec{p}(e)$, output of

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on February 12,2023 at 01:59:57 UTC from IEEE Xplore. Restrictions apply.

the backward pass (from future to past) $\overleftarrow{p}(e)$, and the $h(e)$ output layer itself by reiterating top to the bottom (forwards) from $e=1$ to e_f, bottom to the top (backwards) from $e=e_f$ to 1, and then final values are modified using the following equation:

$$\vec{p}(e) = P(T_{\vec{g}} R_e + V_{\vec{g}} p_{\vec{g}}(e-1) + a_{\vec{g}}) \tag{5}$$

$$\overleftarrow{p}(e) = P(T_{\overleftarrow{g}} R_e + V_{\overleftarrow{g}} p_{\overleftarrow{g}}(e-1) + a_{\overleftarrow{g}}) \tag{6}$$

$$h(m) = P_{\vec{g}} s_{\vec{g}}(e) + O_{\overleftarrow{g}} s_{\overleftarrow{g}}(e) + a_h \tag{7}$$

$$h(e) = \sigma_h(\vec{p}, \overleftarrow{p}). \tag{8}$$

The σ_h function concatenates hidden-layer neurons output sequences and can execute any of these four operations: 1) concatenate; 2) add; 3) average; and 4) multiply. The input to self-attention layer is the sequence of hidden state vectors obtained from BiLSTM, i.e., $h(e)$ [11].

$$\mathcal{M} = \sum_{e=1}^{N} \varrho_e \mathcal{H}_e \tag{9}$$

where ϱ_t is the weighted vector and is evaluated as

$$\varrho_e = \frac{\exp\left(\mathcal{U}_e^{\mathcal{T}} \mathcal{U}_{\mathcal{W}}\right)}{\sum_e \exp\left(\mathcal{U}_e^{\mathcal{T}} \mathcal{U}_{\mathcal{W}}\right)} \tag{10}$$

$$\mathcal{U}_t = \tanh\left(\mathcal{W}_{\mathcal{W}} \mathcal{H}_e + \mathcal{B}_{\mathcal{W}}\right). \tag{11}$$

The *softmax* function at last layer is used to accurately classify threat and normal group. Let $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{\mathcal{T}})$ is the output from attention block and a one-hot encoded $\mathcal{C}$-dimensional vector y denotes the network outcome from the output layer (using a *softmax* function φ). The probability p, which a single input $\mathcal{M}$ corresponds to a certain threat type (y), can be determined as follows:

$$p\left(\widehat{\mathcal{Y}}_c = \mathcal{Y}_c \mid \mathcal{M}\right) = \varphi(\mathcal{M})\, \mathcal{Y}_c = \frac{\exp^{m_c}}{\sum_{d=1}^{\mathcal{C}} \exp^{m_d}}$$
$$(\mathcal{C} = 1, 2, \ldots, c). \tag{12}$$

The $\mathcal{C}$-way cross-entropy loss function gives a probability across $\mathcal{C}$ class labels, which is used to compute the loss for each prediction for all timestamps as follows [11]:

$$\mathcal{LOSS} = \frac{1}{\mathcal{n}} \sum_{i=1}^{n} \sum_{c=1}^{\mathcal{C}} \mathcal{Y}_{ic} \log\left(\widehat{\mathcal{Y}}_{ic}\right) \tag{13}$$

where $\mathcal{n}$ represents the batch size, $\mathcal{C}$ represents the number of classes, and $\mathcal{Y}$ and $\widehat{\mathcal{Y}}$ represent the actual and predicted class labels, respectively.

IV. Security Analysis

The security analysis of the proposed PBDL framework is discussed as follows.

1) Impersonation Attack: An adversary can act as a legitimate $\mathcal{LN}_j$ by sending the following.

1) Sensor temporary identification ($\mathcal{T}_j^{\mathcal{ID}}$).
2) MAC of sensor ($\mathcal{S}_j^{\mathcal{MAC}}$).
3) Geographical location of sensor ($\mathcal{G}_j^{\mathcal{LOC}}$) to $\mathcal{FN}_j$ for creating a temporary id $\mathcal{T}_{kj}$.

Furthermore, timestamp ($\mathcal{T}_{sj}$) is created for request generation of $\mathcal{ID}_j$. However, $\mathcal{FN}_j$ verifies the existing timestamp records. If matches, then it goes for further $\mathcal{ZKP}$ verification and permanent $\mathcal{ID}_j$ creation. If timestamp ($\mathcal{T}_{sj}$) does not matches, then connection gets terminated. Thus, proposed model prevents from impersonation attack.

2) Insider Attack: An adversary can be privileged insider and can get all the information about the $\mathcal{LN}_j$, such as sensor temporary identification ($\mathcal{T}_j^{\mathcal{ID}}$), MAC of sensor ($\mathcal{S}_j^{\mathcal{MAC}}$), geographical location of sensor ($\mathcal{G}_j^{\mathcal{LOC}}$, and timestamp ($\mathcal{T}_{sj}$). However, permanent $\mathcal{ID}_j$ cannot be accessed due to random number generation and salting process. Thus, the model is secure with insider attack.

3) MITM and Replay Attack: An adversary can obtain the message from channel, such as $\mathcal{T}_j^{\mathcal{ID}}$, $\mathcal{S}_j^{\mathcal{MAC}}$, and $\mathcal{G}_j^{\mathcal{LOC}}$, to perform MITM and replay attack. However, with receiving information, $\mathcal{LN}_j$ computes all possible value to find correct $\mathcal{H}[\mathcal{ID}_j]$ using $\mathcal{SALT}$-$\mathcal{P}$ and $\mathcal{T}_{kj}$ combination. The $\mathcal{SALT}$-$\mathcal{Q}$ is evaluated to ensure level of difficulty by appending z value. This value is computed using large prime value ($\mathcal{S}$) and corresponding generator ($\mathcal{T}$), which is difficult to predict. Thus, adversary fails to perform MITM and replay attack.

V. Experimental Results and Evaluation

All experiments were conducted on the Tyrone PC with configuration mentioned in [29]. We have developed the permissioned blockchain scheme using Ethereum and Solidity 6.0 with IPFS version 0.4.19. The DL scheme was developed using TensorFlow library Keras. On the ToN-IoT [30] and IoT-Botnet [31] datasets, the performance of the proposed PBDL for intrusion detection was evaluated. Both datasets were divided into training and testing sets with 70% and 30%, respectively. Finally, as mentioned in [29], feature mapping and normalization were conducted on both datasets. The performance of IDS was measured using four metrics: 1) accuracy; 2) precision; 3) detection rate; and 4) F1-score, mentioned in [7]. The PBDL model was also compared to baseline [i.e., naive bayes (NB), decision tree (DT), and random forest (RF)], standard BiLSTM, and several recently developed state-of-the-art approaches.

A. Results Analysis of Blockchain Scheme

To provide security and privacy in the proposed architecture, each IoT node is first registered in the blockchain scheme. The registration time for numerous IoT nodes is shown in Fig. 2(a). The data upload time of different sensors over the IPFS-secured storage layer is shown in Fig. 2(b), along with the transaction numbers. As the number of transactions grows, so does the upload time. The block mining time, block creation time, and block access time are shown in Fig. 2(c)–(e), respectively. It can be seen that as the number of IoT sensor nodes grows, the time increases, as predicted. Contract deployment time and transaction signing time are shown in Fig. 2(f) and (g), respectively.

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on February 12,2023 at 01:59:57 UTC from IEEE Xplore. Restrictions apply.

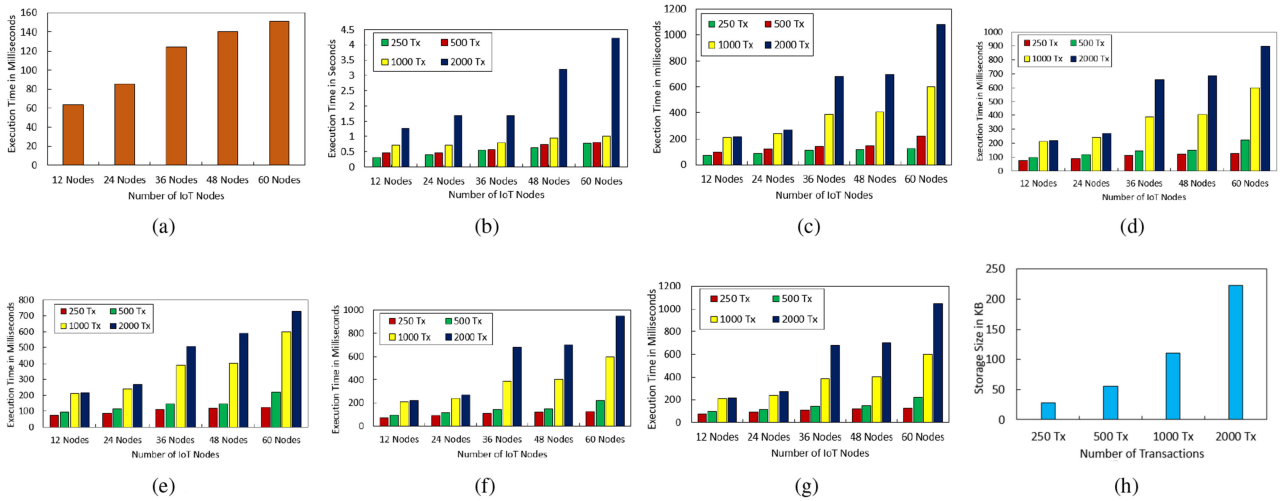

Fig. 2. Results obtained from the blockchain scheme. (a) Analysis of the registration time analysis for IoV nodes. (b) Actual data upload time analysis over the IPFS storage layer. (c) Block mining time analysis for different transactions (*Tx*). (d) Block creation time analysis for different transactions (*Tx*). (e) Analysis of block access time with different transactions (*Tx*). (f) Analysis of contract deployment time with different transactions (*Tx*). (g) Nonrepudiation time analysis with signing of different transaction (*Tx*). (h) Analysis of off-chain layer storage (size in kb) for different transactions.

TABLE IV
COMPARISON OF CLASS-WISE PREDICTION (%) RESULTS WITH TRADITIONAL BILSTM USING THE TON-IOT DATASET

Method	Parameters	Backdoor	DDoS	DoS	Injection	MITM	Normal	Password	Ransomware	Scanning	XSS
BiLSTM	PR	99.73	96.27	99.41	98.38	90.75	99.99	99.96	99.53	99.31	98.18
	DR	99.89	99.90	99.47	99.03	99.64	100.00	98.38	98.73	98.70	97.93
	F1	99.81	97.11	98.99	98.25	91.51	99.99	99.57	99.12	99.32	98.52
	FAR	0.000120	0.001707	0.000264	0.000733	0.000202	0.000062	0.000015	0.000203	0.000309	0.000831
PBDL	PR	99.93	99.03	99.85	99.42	93.42	100.00	99.98	100.00	99.93	99.53
	DR	99.99	97.95	99.58	99.12	99.28	99.99	99.18	99.71	99.33	99.86
	F1	99.91	99.46	99.66	99.23	96.43	100.00	99.68	99.86	99.96	99.73
	FAR	0.000030	0.000446	0.000068	0.000256	0.000144	0.0	0.000007	0.0	0.000030	0.000211

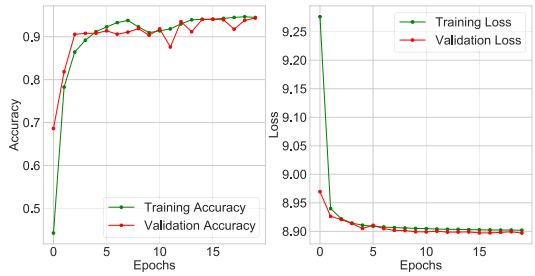

Fig. 3. Accuracy versus loss for the SSVAE technique using the ToN-IoT dataset.

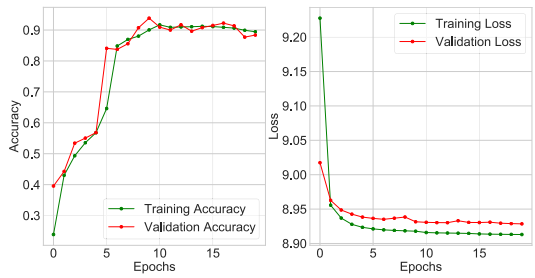

Fig. 4. Accuracy versus loss for the SSVAE technique using the IoT-Botnet dataset.

TABLE V
COMPARISON OF CLASS-WISE PREDICTION (%) RESULTS WITH TRADITIONAL BILSTM USING THE IOT-BOTNET DATASET

Method	Parameters	DoS	DDoS	Reconnaissance	Normal	Theft
BiLSTM	PR	99.62	95.87	96.47	47.25	0.21
	DR	77.81	94.97	99.97	99.79	10.41
	F1	87.38	95.42	98.19	64.13	0.42
	FAR	0.14	1.99	1.45	5.89	1.05
PBDL	PR	99.99	99.99	100.00	99.72	87.17
	DR	99.99	100.00	99.96	99.98	70.83
	F1	99.99	99.99	99.98	99.85	78.16
	FAR	0.000013	0.000006	0.0	0.000144	0.000022

The signature with transactions assures nonrepudiation. The actual storage size in kb increases as the number of transactions increases, as shown in Fig. 2(h).

B. Results Analysis of DL Scheme

The SSVAE approach is used to alter the blockchain scheme's authorized data. The suggested technique is trained and validated using the ToN-IoT and IoT-Botnet datasets. This method is used to avoid inference attacks from being exposed by the learnt model. In both datasets, hyperparameters are initialized using input layer; the *encoder* uses the *two layer*, which includes hidden nodes 50 and 25. As an output layer, *Relu* activation and *sigmoid* are utilized. *Decoder* is made up of a two hidden layer with *hidden nodes* 25 and 50. The final model is configured with *optimizer* $=$ $Nadam$, *loss* $=$ categorical_crossentropy, *batch_size* $= 50$, and *epochs* $= 20$. The results shown in Figs. 3

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on February 12,2023 at 01:59:57 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
COMPARISON OF DR (%) WITH VARIOUS BASELINE TECHNIQUES ON THE TON-IOT DATASET

Techniques	Backdoor	DDoS	DoS	Injection	MITM	Normal	Password	Ransomware	Scanning	XSS
RF	99.98	90.40	91.97	93.53	0.00	100.00	97.81	99.40	95.74	85.47
DT	100.00	100.00	100.00	0.00	0.00	100.00	100.00	100.00	100.00	100.00
NB	99.22	26.80	91.70	92.96	95.11	100.00	75.32	79.98	96.91	19.02
BiLSTM	99.89	99.90	99.47	99.03	99.64	100.00	98.38	98.73	98.70	97.93
PBDL	99.99	97.95	99.58	99.12	99.28	99.99	99.18	99.71	99.33	99.86

TABLE VII
COMPARISON OF DR (%) WITH VARIOUS BASELINE TECHNIQUES ON THE IOT-BOTNET DATASET

Techniques	DoS	DDoS	Reconnaissance	Normal	Theft
RF	99.96	100.00	100.00	14.95	0.00
DT	100.00	100.00	80.06	0.00	100.00
NB	97.76	99.97	81.44	74.76	92.85
BiLSTM	77.81	94.97	99.97	99.79	10.41
PBDL	99.99	100.00	99.96	99.98	70.83

TABLE VIII
COMPARISON OF ACCURACY WITH STATE-OF-THE-ART APPROACHES

Authors	Year	Approach	Dataset	Accuracy
Nguyen et al. [31]	2020	CNN	IoT-Botnet	98.70%
Alsaedi et al. [30]	2020	CART	ToN-IoT	77.00%
Dunn et al. [33]	2021	XGBoost	ToN-IoT	98.00%
Booij et al. [32]	2021	RF	ToN-IoT	98.07%
Poposed Work	2021	PBDL	ToN-IoT	99.89%
			IoT-Botnet	99.98%

Terms and abbreviations: 1) CNN: convolutional neural network; 2) CART: classification and regression trees; and 3) RF: random forest.

and 4 illustrates the efficiency of the SSVAE technique in terms of *acc* versus *loss*. The results report high performance with both datasets, i.e., 94.34% *acc* and 8.89% *loss*, and 88.38% *acc* and 8.92% *loss*, respectively. The SSVAE technique is employed to reshape or convert initial data into new format that can prevent inference attacks. This converted data are then utilized to create a high-performing efficient IDS.

The suggested approach's efficacy as a utility system is also assessed using the SA-BiLSTM model. Both datasets are used to fed input layer with five *hidden layers* and *hidden nodes* = 200, 100, 50, 25, and 15 accordingly, a *Relu* activation function, and a *Softmax* activation function are used to configure the hyperparameters. *loss* = categorical_crossentropy, *optimizer* = $Nadam$, *epochs* = 20, and *batch_size* = 50 are the settings for the final model. The results are based on the existing BiLSTM framework as well as the new PBDL framework. The PBDL with the ToN-IoT dataset obtained 0.0052 *loss* and 99.89 *acc*, whereas the BiLSTM model achieves 99.58 *acc* and 0.0167 *loss*. PBDL with the IoT-Botnet dataset model obtained 0.0685 *loss* and 99.98 *acc*, whereas the BiLSTM model obtained 5.5116 *loss* and 90.86 *acc*.

We also compare and contrast the performance of the proposed PBDL framework with traditional BiLSTM in terms of class-wise prediction outcomes, using PR, DR, F1, and FAR measures. It is reported in Table IV that the PBDL using the ToN-IoT dataset has obtained high numerical values, i.e., an average between 90%–100% for DR, PR, and F1 score, and has achieved 0% FAR. In Table V, the model has obtained high values between 99%–100% PR, DR, and F1 metrics for various types of attacks, such as reconnaissance, DoS, normal group, and DDoS based on the IoT-Botnet dataset; however, with the theft attack model, achieved 70%–87% values. It is seen that the proposed model has increased the performance of traditional BiLSTM.

C. Comparisons With Baseline Approaches

The comparison of PBDL with the baseline approach, such as RF, DT, and NB, and BiLSTM in terms of DR under multiclass classifications (given in Tables VI and VII). The proposed framework can detect different attacks up to 97%–100% in ToN-IoT datasets. Similarly, with the IoT-Botnet dataset, PBDL has outperformed all other competing models and achieved DR between 70%–100%. We conclude that the proposed framework has a higher DR for the majority of attacks and the normal group seen in both datasets.

D. Comparisons With State-of-the-Art Techniques

Table VIII compares the performance of different existing state-of-the-art approaches in terms of accuracy. Alsaedi *et al.* [30], Nguyen *et al.* [31], Booij *et al.* [32], and Gad *et al.* [33] evaluated their work using IoT-Botnet and ToN-IoT datasets. It can be observed that the suggested PBDL framework outperforms existing state-of-the-art approaches by over 1%. The reason for this performance is the combination of permissioned blockchain and SSVAE with SA-BiLSTM. Moreover, the attention mechanism used in the proposed approach has greater impact as it focused only on certain information received from BiLSTM hidden layers that were only required to detect intrusions.

VI. CONCLUSION

A novel framework named PBDL was proposed for industrial healthcare systems to increase ability of data protection as well as to ensure secure data sharing. The permissioned blockchain and smart contract enabled anonymous authentication by implementing a ZKP identification system and prevented data from poisoning attacks. The blockchain solution ensured data with verifiability, nontamper, and transparent features. The off-chain IPFS storage system made PBDL highly scalable with high throughput to access healthcare data. A new DL architecture by combining SSVAE with SA-BiLSTM was also proposed to enforce data privacy (i.e., prevent inference attack) and enhance the attack detection process of the traditional BiLSTM technique. Experiment results on two publicly available datasets proved enhanced performance in terms of detection rate and accuracy over traditional BiLSTM, some baseline, and state-of-the-art approach. Future works include implementation of the PBDL framework with software-defined networks to evaluate the scalability and performance.

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on February 12,2023 at 01:59:57 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] R. K. Dudeja, R. S. Bali, and G. S. Aujla, "Secure and pervasive communication framework using named data networking for connected healthcare," *Comput. Elect. Eng.*, vol. 100, 2022, Art. no. 107806.

[2] J. Xu *et al.*, "Healthchain: A blockchain-based privacy preserving scheme for large-scale health data," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8770–8781, Oct. 2019.

[3] R. Vinayakumar, M. Alazab, S. Srinivasan, Q.-V. Pham, S. K. Padannayil, and K. Simran, "A visualized botnet detection system based deep learning for the Internet of Things networks of smart cities," *IEEE Trans. Ind. Appl.*, vol. 56, no. 4, pp. 4436–4456, Jul./Aug. 2020.

[4] F. Al-Turjman, M. H. Nawaz, and U. D. Ulusar, "Intelligence in the internet of medical things era: A systematic review of current and future trends," *Comput. Commun.*, vol. 150, pp. 644–660, 2020.

[5] G. S. Aujla and A. Jindal, "A decoupled blockchain approach for edge-envisioned IoT-based healthcare monitoring," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 491–499, Feb. 2021.

[6] A. Farouk, A. Alahmadi, S. Ghose, and A. Mashatan, "Blockchain platform for industrial healthcare: Vision and future opportunities," *Comput. Commun.*, vol. 154, pp. 223–235, 2020.

[7] P. Kumar, G. P. Gupta, and R. Tripathi, "An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks," *Comput. Commun.*, vol. 166, pp. 110–124, 2021.

[8] P. Kumar, R. Kumar, G. P. Gupta, R. Tripathi, and G. Srivastava, "P2TIF: A blockchain and deep learning framework for privacy-preserved threat intelligence in industrial IoT," *IEEE Trans. Ind. Informat.*, early access, doi: 10.1109/TII.2022.3142030.

[9] O. Alkadi, N. Moustafa, B. Turnbull, and K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9463–9472, Jun. 2021.

[10] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, and S.-S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Syst.*, vol. 21, no. 1, pp. 49–60, 2015.

[11] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, S. Garg, and M. M. Hassan, "BDTwin: An integrated framework for enhancing security and privacy in cybertwin-driven automotive industrial Internet of Things," *IEEE Internet Things J.*, early access, doi: 10.1109/JIOT.2021.3122021.

[12] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, T. R. Gadekallu, and G. Srivastava, "SP2F: A secured privacy-preserving framework for smart agricultural unmanned aerial vehicles," *Comput. Netw.*, vol. 187, 2021, Art. no. 107819.

[13] Y. Yu, Y. Li, J. Tian, and J. Liu, "Blockchain-based solutions to security and privacy issues in the Internet of Things," *IEEE Wireless Commun.*, vol. 25, no. 6, pp. 12–18, Dec. 2018.

[14] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[15] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 2002, pp. 337–351.

[16] L. Zhang, Y. Zou, W. Wang, Z. Jin, Y. Su, and H. Chen, "Resource allocation and trust computing for blockchain-enabled edge computing system," *Comput. Secur.*, vol. 105, 2021, Art. no. 102249.

[17] D. Liu, Y. Zhang, W. Wang, K. Dev, and S. A. Khowaja, "Flexible data integrity checking with original data recovery in IoT-enabled maritime transportation systems," *IEEE Trans. Intell. Transp. Syst.*, early access, doi: 10.1109/TITS.2021.3125070.

[18] A. Tandon, A. Dhir, N. Islam, and M. Mäntymäki, "Blockchain in healthcare: A systematic literature review, synthesizing framework and future research agenda," *Comput. Ind.*, vol. 122, 2020, Art. no. 103290.

[19] R. Gupta, S. Tanwar, F. Al-Turjman, P. Italiya, A. Nauman, and S. W. Kim, "Smart contract privacy protection using ai in cyber-physical systems: Tools, techniques and challenges," *IEEE Access*, vol. 8, pp. 24746–24772, 2020.

[20] H.-N. Dai, M. Imran, and N. Haider, "Blockchain-enabled internet of medical things to combat COVID-19," *IEEE Internet Things Mag.*, vol. 3, no. 3, pp. 52–57, Sep. 2020.

[21] R. Kumar and R. Tripathi, "Towards design and implementation of security and privacy framework for Internet of Medical Things (IoMT) by leveraging blockchain and IPFS technology," *J. Supercomputing*, vol. 77, no. 8, pp. 7916–7955, 2021.

[22] M. A. Rahman, M. S. Hossain, M. S. Islam, N. A. Alrajeh, and G. Muhammad, "Secure and provenance enhanced Internet of Health Things framework: A blockchain managed federated learning approach," *IEEE Access*, vol. 8, pp. 205071–205087, 2020.

[23] P. Kumar, G. P. Gupta, and R. Tripathi, "TP2SF: A trustworthy privacy-preserving secured framework for sustainable smart cities by leveraging blockchain and machine learning," *J. Syst. Architecture*, vol. 115, 2020, Art. no. 101954.

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

[25] M. Begli, F. Derakhshan, and H. Karimipour, "A layered intrusion detection system for critical infrastructure using machine learning," in *Proc. IEEE 7th Int. Conf. Smart Energy Grid Eng.*, 2019, pp. 120–124.

[26] R. M. S. Priya *et al.*, "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," *Comput. Commun.*, vol. 160, pp. 139–149, 2020.

[27] A. I. Newaz, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, "Healthguard: A machine learning-based security framework for smart healthcare systems," in *Proc. 6th Int. Conf. Social Netw. Anal., Manage. Secur.*, 2019, pp. 389–396.

[28] D. He *et al.*, "Intrusion detection based on stacked autoencoder for connected healthcare systems," *IEEE Netw.*, vol. 33, no. 6, pp. 64–69, Nov./Dec. 2019.

[29] P. Kumar, R. Kumar, G. P. Gupta, and R. Tripathi, "A distributed framework for detecting DDoS attacks in smart contract-based blockchain-IoT systems by leveraging fog computing," *Trans. Emerg. Telecommun. Technol.*, vol. 36, no. 6, 2021, Art. no. e4112.

[30] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020.

[31] H.-T. Nguyen, Q.-D. Ngo, and V.-H. Le, "A novel graph-based approach for IoT botnet detection," *Int. J. Inf. Secur.*, vol. 19, no. 5, pp. 567–577, 2020.

[32] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, "ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion datasets," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 485–496, Jan. 2022.

[33] A. R. Gad, A. A. Nashat, and T. M. Barkat, "Intrusion detection system using machine learning for vehicular ad hoc networks based on ton-IoT dataset," *IEEE Access*, vol. 9, pp. 142206–142217, 2021.

Authorized licensed use limited to: Southern Illinois University Carbondale. Downloaded on February 12,2023 at 01:59:57 UTC from IEEE Xplore. Restrictions apply.