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

Kumar Randhir, Kumar Prabhat, Tripathi Rakesh, Gupta Govind P., Islam A. K. M. Najmul, Shorfuzzaman Mohammad

DOI: 10.1109/TII.2022.3161631

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

Randhir Kumar, *Member*, *IEEE*, Prabhat Kumar, *Student Member*, *IEEE*, 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 smart contract-based consensus mechanism. Second, the authenticated data is 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 attack detection process. The security analysis and experimental results using IoT-Botnet and ToN-IoT datasets confirms the superiority of PBDL framework over existing state-of-the-art techniques.**

***Index Terms*—Blockchain, Deep-Learning, Healthcare Systems, Industrial Internet of Things (IIoT), Intrusion Detection System, Privacy-Preservation**

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 Internet of Things (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 takes the advantage from IIoT where industrial sensors and actuators are used as wearable devices to collect users physiological data, such as blood pressure, electrocardiogram (ECG), 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, 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 another major challenges in the present industrial healthcare systems. We believe data privacy is mostly related to Active Data Privacy Attacks (ADPA) and Passive Data Privacy Attacks (PDPA) [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 patients health data. Moreover, it can negatively impact the performance of artificial intelligence-based data analytic or attack detection process of Intrusion Detection Systems (IDS) [10]. On the other hand, PDPA is launched by the attacker to sniff (private) data i.e., to gain some fundamental statistical properties from training dataset (such as data inference attacks) [11]. Moreover, privacy breaches are also related to authentication i.e., a condition where an 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, that can be used to minimize authentication related privacy breaches [13].

A. Threat Model

The widely adopted "Dolev-Yao (DY) threat model" [14] is used in designing the proposed PBDL framework. According

(Corresponding Authors: Prabhat Kumar and Govind P. Gupta)

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 (Reference No. MTR/2019/001285) and in part by the Taif University Researchers Supporting Project number (TURSP-2020/79), Taif University, Taif, Saudi Arabia.

Randhir Kumar, Prabhat Kumar, Rakesh Tripathi and Govind P. Gupta are all 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)

to "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 "DY model", the current de facto "Canetti and Krawczyk's adversary model", [15] known as the "CKadversary model" is also utilized in designing 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 "DY model", in "CKadversary model" edge and cloud nodes are considered as semi-trusted entities and registration authority is assumed as the trusted entity in the network [8].

B. Key Contribution

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

- Permissioned blockchain and smart contracts are combined with deep learning 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.
- In PBDL, a blockchain scheme is designed that first registers the participating entities, then verifies them using Zero Knowledge Proof (ZKP) identification system and finally validates using 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.
- The authenticated data is used by the proposed DL scheme. The latter combines SSVAE with self-attention based-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 is further used by Self-Attention based-BiLSTM (SA-BiLSTM) technique for intrusion detection. We employed attention mechanism to concentrate more on the information extracted from the forward and backward hidden layers of BiLSTM.

The rest of the paper 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 and conclusion with future directions is presented in Section V and Section VI, respectively.

II. Related Work

In order to overcome the aforementioned challenges, various solution related to blockchain have been proposed in literature [16], [17]. For example, Tandon et al. [18] highlighted the importance of security, and privacy in healthcare system, and suggested the advantage and challenges of utilizing blockchain as a solution in healthcare system. Farouk et al. [6] illustrated the need of data privacy protection in IoT-enabled healthcare system and emphasized on how blockchain technology are used to achieve privacy goals. Turjman et al. [4] discussed different ways to integrate blockchain with healthcare system in order to address issues like 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 research [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 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 performs 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 Deep Learning (DL)-based techniques to design IDS in healthcare systems. For instance, Kumar et al. [7] designed an IDS based on fog-cloud architecture and ensemble learning in healthcare environment. Begli et al. [25] proposed a secure IDS (SVM-IDS) in remote healthcare systems. Furthermore, Swarna et al. [26] presented a Deep Neural Network (DNN)-based IDS for 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. The proposed framework

A. Overall Systematic Architecture

The systematic architecture of proposed PBDL framework is made up of three layers, namely, *(i)* Industrial healthcare system layer, *(ii)* Edge-Blockchain layer and *(iii)* Cloud-Blockchain layer, as shown in Fig 1.

(i) *Industrial healthcare system layer:* In this layer various IoT-based healthcare tracking systems and implantable medical devices (e.g., temperature sensors, glucose monitor, 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 node* (LN).

(ii) *Edge-Blockchain layer:* The layer is made up of powerful nodes, such as data analysis servers, industrial computers, edge-computing servers, and so on, and are referred as *full nodes* (FN). The peer-to-peer network is built through geo-distributed regions of edge devices located in primary and

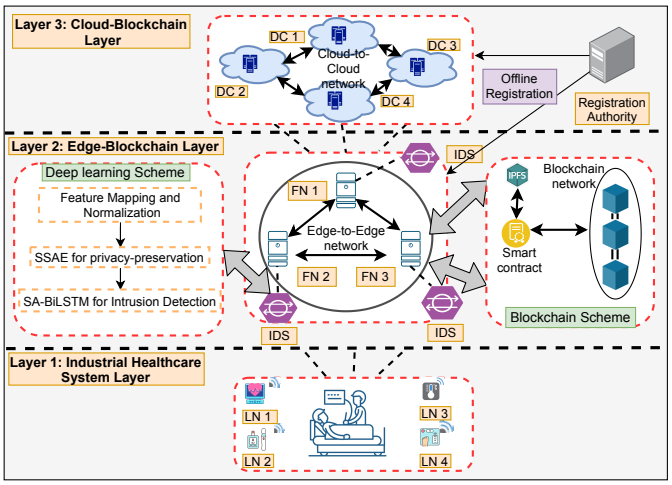


Fig. 1: System Model

urban health centres. Each patient (i.e., associated healthcare device) is assigned with edge service node, that collects, process, and raise alarms in emergency situations, and interacts with cloud for long-term storage and backup.

(iii) *Cloud-Blockchain layer:* This layer includes a number of cloud providers or vendors and data centres. These Data Centres (DCs) are in charge of providing clients with services (like computation, processing, and so on).

In a typical healthcare system, we mainly have LNs, FNs, and DCs as communicating entities. In such scenario, LNs have limited resources and can only communicate data to FNs over edge-blockchain layer. The FNs assists 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 is 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 ZKP protocol and finally, the smart contract-based consensus enhanced Proof of Work (ePoW) 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 (SaaS) (i.e., edge servers and cloud data centres, gateways, 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 cyber-attacks.

B. Blockchain Scheme

In the proposed PBDL framework, a permissioned blockchain is designed due to two principal reasons. 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. Second, permissionless blockchain are open and can increase attacks from external adversaries without a notable function enhancement. Therefore, in our opinion are unsuitable in industrial healthcare systems. The proposed scheme has four different phases namely (i) system registration, (ii) verification using Zero Knowledge Proof ($\mathcal{ZKP}$), (iii) validation and block creation phase, and (iv) data generation and block updation phase. We have discussed the steps and working of each phase below:

(i) *Registration Phase:* In this step, a trusted registration authority registers the data centre ($\mathcal{DC}_j$) and full node ($\mathcal{FN}_j$) safely in off-line mode. In addition, using the zero knowledge proof ($\mathcal{ZKP}$) 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 zero knowledge proof 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:

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 i.e., (i) sensor temporary identification of $\mathcal{LN}_j$ ($\mathcal{TID}_j$) (ii) MAC of sensor ($\mathcal{S}_j^{\mathcal{MAC}}$), and (iii) geographical location of sensor ($\mathcal{G}_j^{\mathcal{LOC}}$). It is worth mentioning here that in 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. Further, 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 secure channel.

Step 2: Once ($\mathcal{FN}_j$) receives $\mathcal{T}_{kj}$, it extracts the included parameters. Then $\mathcal{SALT}$-$\mathcal{P}$ which generates pseudo random number that gets appended to $\mathcal{T}_{kj}$, to protect from pre-computed hash attack. To create unique hash a SALT adds random bits before hash computation, that prevents from attack of pre-computed hash. Further, the $\mathcal{SALT}$-$\mathcal{P}$ 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 secure channel.

(ii) *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 $\mathcal{H}[\mathcal{ID}_j]$ utilising $\mathcal{SALT}$-$\mathcal{P}$ 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 (($\mathcal{T}_{sj}$)), as y= $\mathrm{T}^{\mathcal{T}_{sj}}$ *mod* S. But, value of ($\mathcal{T}_{sj}$) can not be revealed in the complete process. As, the complete process is highly depends on the understanding of ($\mathcal{T}_{sj}$) without any disclose. The $\mathcal{LN}_j$ finds random number (u) that is further used in the computation of v= T^{u} *mod* S. The d value is applied to create $\mathcal{H}_i$ by $\mathcal{SALT}$-$\mathcal{Q}$ and

TABLE I: Registration and verification of $\mathcal{LN}_j$

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

TABLE II: The process of block creation and validation

Light Node ($\mathcal{LN}_j$)	Full Node ($\mathcal{FN}_j$)	Full Node ($\mathcal{FN}_j$) peer
INPUT: $\mathcal{IDD}_j$ OUTPUT: B_j Create key pair (PB_{kj},PR_{kj}) of $\mathcal{IDD}_j$ PB`kj ⟶ (SSL/TLS)		
	Create $\mathcal{sig}_j$ sig`j ⟵ (SSL/TLS)	
Validate ($\mathcal{sig}_j$) Send (PB_{kj},$\mathcal{G}_j^{\mathcal{LOC}}$) (Request to join PB`kj,G`LOC`j, ID`j) ⟶ (SSL/TLS)		
	Send $\mathcal{G}_j^{\mathcal{LOC}}$ to $\mathcal{FN}_j$ peer nodes G`LOC`j ⟶ (SSL/TLS)	
		Check $\mathcal{G}_j^{\mathcal{LOC}}$ with timestamp, latitude, and longitude of $\mathcal{LN}_j$ Return status (True/False) (True) ⟵ (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) ⟶ (SSL/TLS)	

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: $\mathcal{Data}_j$ OUTPUT: Update B_j read ($\mathcal{Data}_j$) which includes $\mathcal{S}_j^{\mathcal{MAC}}$, PB_{kj}, $\mathcal{IDD}_j$, and $\mathcal{G}_j^{\mathcal{LOC}}$ of $\mathcal{LN}_j$ Sign ($\mathcal{sig}_j$)=($\mathcal{Data}_j$, PR_{kj}) Create T_j= ($\mathcal{Data}_j$, PB_{kj}, $\mathcal{sig}_j$,$\mathcal{IDD}_j$) Send T_j (T`j) ⟶ (SSL/TLS)		
	Validate $\mathcal{IDD}_j$, PB_{kj} Check $\mathcal{Data}_j$, $\mathcal{sig}_j$ Validate $\mathcal{sig}_j$ Add T_j to B_j Disseminate to $\mathcal{FN}_j$ peer (B`j) ⟶ (SSL/TLS)	
		Validate B_j Synchronized Blockchain

sent to ($\mathcal{FN}_j$). Further, 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 $\mathcal{SALT}\text{-}\mathcal{Q}$ 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 $\mathcal{SALT}\text{-}\mathcal{Q}$ for transmit to $\mathcal{LN}_j$. Once the $\mathcal{LN}_j$ receives the q_1 then it computes answer A_1 and appends the $\mathcal{SALT}\text{-}\mathcal{Q}$ 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 ($\mathcal{IDD}_j$) to $\mathcal{LN}_j$. The same $\mathcal{IDD}_j$ is disseminated to the blockchain network for further communication. The entire process of registration and verification is summarized in TABLE I.

(iii) *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 shown 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}, PR_{kj}), with PB_{kj} being the public key whereas PR_{kj} being the private key of the j^{th} light node ($\mathcal{LN}_j$). In addition, $\mathcal{FN}_j$ starts the registration process.

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

Step 3: The $\mathcal{LN}_j$ validates a signature. When the $\mathcal{sig}_j$ 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 $\mathcal{IDD}_j$.

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 $\mathcal{IDD}_j$ for True status.

(iv) *Data creation and updation of Block :* This phase includes data generation by $\mathcal{LN}_j$. The TABLE III shows summarized view of data generation and updation of block. The created data is identified as transactions (T_j). The entire process of data creation and alteration in a block is discussed below.

Step 1: At first step, data ($\mathcal{Data}_j$) is generated by $\mathcal{S}_j^{\mathcal{MAC}}$ and signed ($\mathcal{sig}_j$) with PR_{kj} of $\mathcal{LN}_j$, after successful signa-

ture of $\mathcal{D}ata_j$ a new transaction (T_j) gets generated including the credential $\mathcal{S}ig_j$, PB_{kj}, $\mathcal{ID}_j$ of $\mathcal{LN}_j$. Further, T_j gets forwarded to $\mathcal{FN}_j$ for its validation and updation in B_j.

Step 2: Further, PB_{kj} gets associated with $\mathcal{ID}_j$ and valid record gets verified with credential $\mathcal{D}ata_j$ and $\mathcal{S}ig_j$. 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. Deep-Learning 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], [23]. Then we design a Stacked Sparse Variational AutoEncoder (SSVAE) technique to reshape or encode data (used to prevent inference attacks) and the encoded data is finally used by the proposed Self-Attention-based Bidirectional Long Short Term Memory (SA-BiLSTM) for intrusion detection.

The VAE technique, works on the principal of graphical model with 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.

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

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.

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

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:

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

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 side of Eq.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 mentioned below.

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

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 Stacked SVAE (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.

Further to verify the effectiveness of SSVAE based privacy-preservation approach, a utility system based on Self-Attention-based BiLSTM (SA-BiLSTM) is designed to retain data privacy. The traditional BiLSTM initially originated from 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 input datasets. BiLSTMs offers 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 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 below equation:

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

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

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

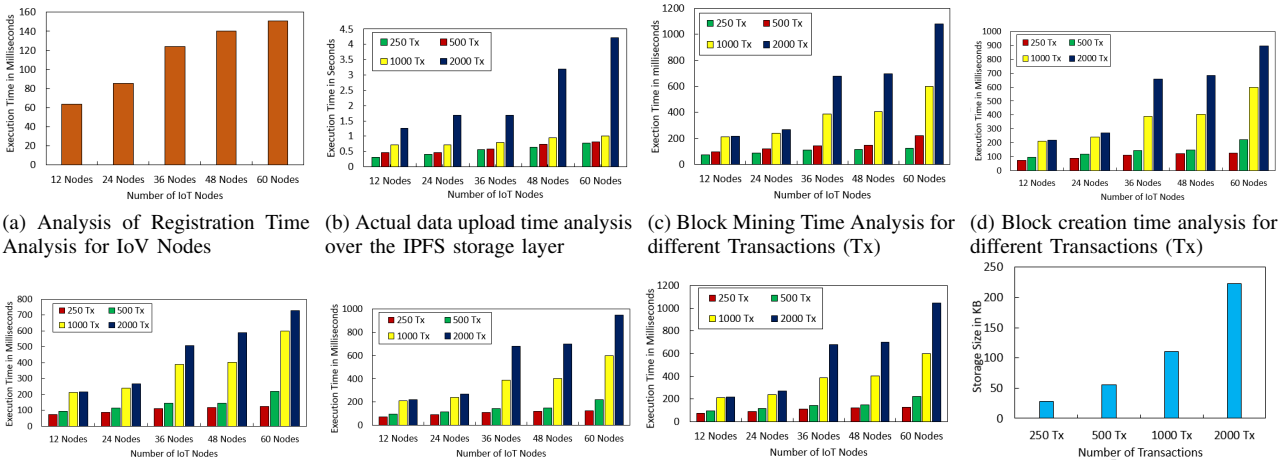
$$h(e) = \sigma_h(\vec{p}, \overleftarrow{p}) \tag{8}$$

The σ_h function concatenates hidden layer neurons output sequences and can execute any of these four operations: concatenate, add, average, and 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 (weighted vector) is evaluated as;

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



(a) Analysis of 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) Non-Repudiation Time Analysis with Signing of different Transaction (Tx)

(h) Anaysis of Off-chain layer storage (Size in KB) for different Transactions

Fig. 2: Results obtained from blockchain scheme

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

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 that a single input $\mathcal{M}$ corresponds to a certain threat type (y) can be determined as follows:

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

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, $\mathcal{Y}$ 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 below.

Impersonation Attack: An adversary can act as a legitimate $\mathcal{LN}_j$ by sending the (i) sensor temporary identification ($\mathcal{T}_j^{\mathcal{ID}}$) (ii) MAC of sensor ($\mathcal{S}_j^{\mathcal{MAC}}$), and (iii) geographical location of sensor ($\mathcal{G}_j^{\mathcal{LOC}}$) to $\mathcal{FN}_j$ for creating a temporary id $\mathcal{T}_{\mathcal{k}j}$. Further,timestamp ($\mathcal{T}_{\mathcal{s}j}$) 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 ($\mathcal{T}_{\mathcal{s}j}$) does not matches then, connection gets terminated. Thus, proposed model prevents from impersonation attack.

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

MITM and Replay attack: An adversary can obtain the message from channel like $\mathcal{T}_j^{\mathcal{ID}}$, $\mathcal{S}_j^{\mathcal{MAC}}$, 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 $\mathcal{T}_{\mathcal{k}j}$ combination. The $\mathcal{SALT}$-$\mathcal{Q}$ is evaluated to ensures 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 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: accuracy, precision, detection rate and 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.

TABLE IV: Comparison of class wise prediction (%) results with traditional BiLSTM using 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

TABLE V: Comparison of class wise prediction (%) results with traditional BiLSTM using 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

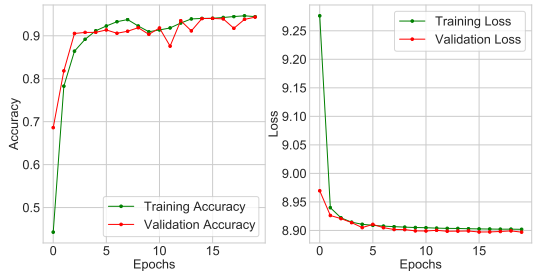


Fig. 3: The Accuracy vs loss for SSVAE technique using ToN-IoT dataset

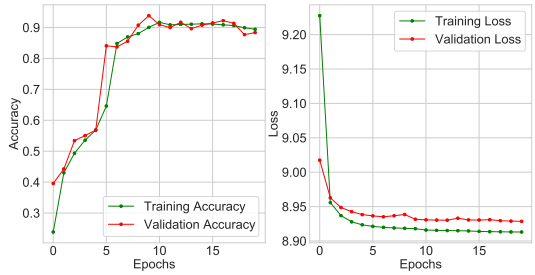


Fig. 4: The Accuracy vs loss for SSVAE technique using IoT-Botnet dataset

The registration time for numerous IoT nodes is shown in 2a. The data upload time of different sensors over the IPFS secured storage layer is shown in Fig. 2b, 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 depicted in 2c, 2d, and 2e. 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 depicted in 2f and 2g, respectively. the signature with transactions assures non-repudiation. The actual storage size in KB increases as the number of transactions increases, as seen in Fig. 2h.

B. Results analysis of deep-learning 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, hyper parameters are initialized using input layer; the *encoder* uses the *2 layer*, which includes hidden nodes *50 and 25*. As an output layer, *Relu* activation and *sigmoid* are utilized. *Decoder* is made up of a *2* hidden layer with *hidden nodes 25, 50*. The final model is configured with *optimizer*= $Nadam$, *loss*= $categorical\_crossentropy$, *batch_size*= 50 and *epochs*=20. The results shown in Fig. 3 and Fig. 4 illustrates the efficiency of SSVAE technique in terms of *acc* vs *loss*. The results reports 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 is 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 5 *hidden layers*, and *hidden nodes*= 200, 100, 50, 25, 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 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 IoT-Botnet dataset model obtained 0.0685 *loss* and 99.98 *acc*, whereas 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 IoT-Botnet dataset. However, with 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 baseline approach such as RF, DT, and NB and BiLSTM in terms of DR under multiclass classifications (shown in Table VI and Table VII). The proposed framework can detect different attacks up to $97\% - 100\%$ in ToN-IoT datasets. Similarly, with IoT-Botnet dataset, PBDL has outperformed all other competing models

TABLE VI: Comparison of DR (%) with various baseline techniques on 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 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 & Abbreviations: CNN: Convolutional Neural Network, CART: Classification and Regression Trees, RF: Random Forest.

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. The work published in [31], [30], [32], [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, non-tamper 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 attack detection process of traditional BiLSTM technique. Experiment results on two publicly available datasets proves 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 scalibility and performance.

References

[1] R. K. Dudeja, R. S. Bali, and G. S. Aujla, "Secure and pervasive communication framework using named data networking for connected healthcare," *Computers & Electrical Engineering*, vol. 100, p. 107806, 2022.

[2] J. Xu, K. Xue, S. Li, H. Tian, J. Hong, P. Hong, and N. Yu, "Healthchain: A blockchain-based privacy preserving scheme for large-scale health data," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8770–8781, 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 Transactions on Industry Applications*, vol. 56, no. 4, pp. 4436–4456, 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," *Computer Communications*, 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 Journal on Selected Areas in Communications*, pp. 1–1, 2020.

[6] A. Farouk, A. Alahmadi, S. Ghose, and A. Mashatan, "Blockchain platform for industrial healthcare: Vision and future opportunities," *Computer Communications*, 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," *Computer Communications*, 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 Transactions on Industrial Informatics*, pp. 1–1, 2022.

[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 of Things Journal*, pp. 1–1, 2020.

[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 Systems*, 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 of Things Journal*, pp. 1–1, 2021.

[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," *Computer Networks*, vol. 187, p. 107819, 2021.

[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 Communications*, vol. 25, no. 6, pp. 12–18, 2018.

[14] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[15] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 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," *Computers & Security*, vol. 105, p. 102249, 2021.

[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 Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.

[18] A. Tandon, A. Dhir, N. Islam, and M. Mäntymäki, "Blockchain in healthcare: A systematic literature review, synthesizing framework and future research agenda," *Computers in Industry*, vol. 122, p. 103290, 2020.

[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. 24 746–24 772, 2020.

[20] H.-N. Dai, M. Imran, and N. Haider, "Blockchain-enabled internet of medical things to combat covid-19," *IEEE Internet of Things Magazine*, vol. 3, no. 3, pp. 52–57, 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," *The Journal of 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. 205 071–205 087, 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," *Journal of Systems Architecture*, p. 101954, 2020.

[24] T. A. Adesuyi and B. M. Kim, "A layer-wise perturbation based privacy preserving deep neural networks," in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2019, pp. 389–394.

[25] M. Begli, F. Derakhshan, and H. Karimipour, "A layered intrusion detection system for critical infrastructure using machine learning," in *2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE)*. IEEE, 2019, pp. 120–124.

[26] S. P. R.M., P. K. R. Maddikunta, P. M., S. Koppu, T. R. Gadekallu, C. L. Chowdhary, and M. Alazab, "An effective feature engineering for dnn using hybrid pca-gwo for intrusion detection in iomt architecture," *Computer Communications*, 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 *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2019, pp. 389–396.

[28] D. He, Q. Qiao, Y. Gao, J. Zheng, S. Chan, J. Li, and N. Guizani, "Intrusion detection based on stacked autoencoder for connected healthcare systems," *IEEE Network*, vol. 33, no. 6, pp. 64–69, 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," *Transactions on Emerging Telecommunications Technologies*, vol. n/a, no. n/a, p. 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. 165 130–165 150, 2020.

[31] H.-T. Nguyen, Q.-D. Ngo, and V.-H. Le, "A novel graph-based approach for iot botnet detection," *International Journal of Information Security*, 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 of Things Journal*, pp. 1–1, 2021.

[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*, 2021.

Randhir Kumar is working towards the Ph.D. degree in Department of Information Technology, National Institute of Technology, Raipur. He has published his research article in leading journal and conferences from IEEE, Elsevier, Springer, and John Wiley and has authored or coauthored over 35 publications. His research interest includes cryptographic techniques, information security, blockchain technology, and web mining. He is also an IEEE Member.

Prabhat Kumar is working towards his Ph.D. degree in Information Technology, National Institute of Technology, Raipur, India. He earned his Ph.D. scholarship position as talented student. His research interests are Security and Privacy of the Internet of Things, artificial intelligence, software-defined networking, Cybersecurity, and Blockchain. He has authored or coauthored over 20 publications in high-ranked journals and conferences.

Rakesh Tripathi received his Ph.D. degree in Computer Science and Engineering from the Indian Institute of Technology Guwahati, India. He is an Assistant Professor with the Department of Information Technology, National Institute of Technology, Raipur, India. He has authored or coauthored over 50 publications in high-ranked journals and conferences. His research interests include Distributed Systems, Intrusion Detection System, IoT and Blockchain. Some of his research findings are published in top cited journals, such as IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT (IEEE TNSM), IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS (TII), IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (TITS), IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING (TNSE), IEEE MICRO, and IEEE IoTJ. He is also an IEEE Senior Member.

Govind P. Gupta received his Ph.D. degree from Indian Institute of Technology, Roorkee, India, in 2014. He is currently an Assistant Professor in the Department of Information Technology at National Institute of Technology, Raipur, India. He has authored or coauthored over 70 publications in high-ranked journals and conferences. Some of his research findings are published in top cited journals, such as IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS (TII), IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING (TNSE), IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (TITS), IEEE IoTJ and IEEE MICRO. His current research interests include efficient protocol design for Wireless Sensor Networks and Internet of Things, Network Security and Software-defined Networking. He is a professional member of the IEEE and ACM.

A.K.M. Najmul Islam is an Associate Professor at LUT University, Finland. He conducts cross-disciplinary research in the area of digitalization and its impact on citizens, organizations, and society. He is a docent of Information Systems at Tampere University. Islam's publication has appeared in top Information Systems outlets such as Journal of Strategic Information Systems, European Journal of Information Systems and Information Systems Journal. He has published in other highly ranked interdisciplinary journals such as IEEE Access, Computers & Education, Technological Forecasting and Social Change, International Journal of Information Management, Information Technology & People, Computers in Human Behavior, Computers in Industry, Internet Research, Communications of the AIS, among others. He is currently serving as a Senior Editor for Information Technology & People journal.

Mohammad Shorfuzzaman is currently an Associate Professor with the Department of Computer Science, College of Computers and Information Technology (CCIT), Taif University, Taif, Saudi Arabia. He is also a member of the Big Data Analytics and Applications (BDAAG) Research Group, CCIT. His current research interests include applied artificial intelligence in the areas of computer vision, natural language processing, big data, and cloud computing.