BinDaaS: Blockchain-Based Deep-Learning as-a-Service in Healthcare 4.0 Applications

Pronaya Bhattacharya, Sudeep Tanwar, Umesh Bodkhe, Sudhanshu Tyagi, *Senior Member, IEEE*, and Neeraj Kumar, *Senior Member, IEEE*

***Abstract*—Electronic Health Records (EHRs) allows patients to control, share, and manage their health records among family members, friends, and healthcare service providers using an open channel, i.e., Internet. Thus, privacy, confidentiality, and data consistency are major challenges in such an environment. Although, cloud-based EHRs addresses the aforementioned discussions, but these are prone to various malicious attacks, trust management, and non-repudiation among servers. Hence, blockchain-based EHR systems are most popular to create the trust, security, and privacy among healthcare users. Motivated from the aforementioned discussions, we proposes a framework called as Blockchain-Based Deep Learning as-a-Service (BinDaaS). It integrates blockchain and deep-learning techniques for sharing the EHR records among multiple healthcare users and operates in two phases. In the first phase, an authentication and signature scheme is proposed based on lattices-based cryptography to resist collusion attacks among *N-1* healthcare authorities from N. In the second phase, Deep Learning as-a-Service (DaaS) is used on stored EHR datasets to predict future diseases based on current indicators and features of patient. The obtained results are compared using various parameters such as accuracy, end-to-end latency, mining time, and computation and communication costs in comparison to the existing state-of-the-art proposals. From the results obtained, it is inferred that BinDaaS outperforms the other existing proposals with respect to the aforementioned parameters.**

***Index Terms*—Authentication, Blockchain, Deep-Learning, EHRs.**

I. Introduction

STORAGE of health records of patients has changed over the years from manual to electronic form. To support this change, and to induce new systems in operations and maintenance of health records, the healthcare industries also saw a transition from Healthcare *1.0* to Healthcare *4.0*. In Healthcare *1.0*, industries are more centric towards manual paper records with fixed number of patient visit cycles to clinics. The records were subjected to wear-and-tear over a period of time, and patient privacy and confidentiality was at high risk. To ensure better maintenance and scalability, Healthcare *2.0*, known as *e-Health* was introduced in early $2000's$ that focused on technological advancements using devices and reduced human effort, hence was more data-driven to increase user experience through cloud based servers. However, cloud servers were prone to active security attacks by malicious entities to gain access to the patient critical data which may be sold or used for personal purposes. To overcome this, the focus has shifted towards decentralizing healthcare applications, known as Healthcare *3.0*. It focused on usage of mobile apps to store EHR to have cost saving and efficient management. The apps still lacked intelligence to allow personalized health services for patients. With the advent of artificial intelligence (AI) and Internet-of-Things (IoT) in healthcare, distributed EHR records became more intelligent to support real-time analytics and medical tracking, remote monitoring, emergency services, and mobility through smart wearables. The wearables communicate via networking infrastructures like ZWave, Bluetooth, 3 G, ZigBee, and 4 G (LTE). AI empowers decision models and health recommendations to support business intelligence. However, in healthcare *4.0*, the issues of data fragmentation, complex learning models and lack of interoperability among various stakeholders like patients, doctors, medics, hospitals, and logistics needs to be addressed.

To address the aforementioned issues pertaining to healthcare 4.0 applications, a blockchain-based EHR framework can be used, that provides a trust and interoperability among all the stakeholders. It has a distributed, auditable, immutable, chronological and timestamped ledger to store medical data. Blockchain has a large number of applications ranging from finance [1], education [2], edge computing services [3], tourism [4], automation [5], and many more. In healthcare *4.0*, the amount of data has increased two-folds having $50 billion market [6]. The amount of certified EHRs has also drastically doubled from 42% to 87% [7]. This humongous amount of data generated from various sources is unstructured because of frequent updation in records made by doctors, hospitals, medics, and other service providers. It leads to maintenance

Manuscript received September 5, 2019; revised December 16, 2019; accepted December 20, 2019. Date of publication December 25, 2019; date of current version July 7, 2021. Recommended for acceptance by Dr. Y. Wu. *(Corresponding author: Neeraj Kumar.)*

P. Bhattacharya, S. Tanwar, and U. Bodkhe are with the Department of Computer Science and Engineering, Institute of Technology, Nirma University Ahmedabad 382481, India (e-mail: pronoya.bhattacharya@nirmauni.ac.in; sudeep.tanwar@nirmauni.ac.in; umesh.bodkhe@nirmauni.ac.in).

S. Tyagi is with the Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Patiala 147001, India (e-mail: sudhanshutyagi123@gmail.com).

N. Kumar is with the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala 147001, India, with the King Abdulaziz University, Jeddah 21589, Saudi Arabia, and also with the Department of Computer Science and Information Engineering, Asia University, Taichung City 41354, Taiwan (e-mail: neeraj.kumar@thapar.edu).

Digital Object Identifier 10.1109/TNSE.2019.2961932

2327-4697 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

issues such as claim settlements, bills management, and drug tracking. According to guidelines issued by Health Insurance Portability and Accountability Act (HIPPA) [8]–[10] stakeholders in healthcare scenario should ensure authorized data, with users themselves uploading certified EHRs by doctors. The challenges for Blockchain adaptation in healthcare are-data ownership, privacy concerns of patient sensitive data, quantum and collusion attacks, scalability of mined transactions, storage, capabilities, and the cost of maintaining healthcare blockchain. To address limitations of user privacy, data ownership, and to resist quantum and collusion attacks, lattice-based cryptography provides an ideal solution. Lattices form post-quantum blockchain networks (P-QBN) which are found to be secure in the random oracle model [11]. Thus, certified EHRs are authorised by lattice signature generation and verification operations. This ensures perfect secrecy of records, as per the HIPPA guidelines.

To address the limitations of scalability of mined transactions, cost and storage issues of EHR across distributed nodes in blockchain, distributed AI can empower blockchain operations [12]. In the similar direction, earlier traditional Machine Learning (ML) and statistical methods [13] were also used, but they analyzed from raw data to select appropriate features and form patterns of prior interest. It is a skilled art to select parameters of interest which is more time-consuming [14], whereas the Deep-Learning (DL) techniques, learn about feature selection from data itself and then without any human intervention, allows the discovery of hidden complex relationships among the data.

Many researchers have provided efficient solutions to address security and privacy concerns for storing EHR in blockchain. For example, Christian *et al.* [15] proposed a conceptual framework to store EHR records over the blockchain to address the privacy concerns of the patient. The exact internal structure and format to store EHRs were not discussed. Lanxiang *et al.* [16] proposed logic expressions for calculating indexes in EHR which was further migrated to blockchain network. But, it has overheads related to index searching in blockchain which increases with more transactions, hence not scalable. Amin *et al.* [17] designed a medical health architecture with mutual authentication for single server, and allow anonymity and resilience at lower communication cost. The trust issues of sever-client interactions were not discussed. Gope *et al.* [18] presented an IoT-based secure and anonymous RFID tag structure at low latency, to deal with impersonation attacks. The identity of the server can be compromised in the existing scheme. Peng *et al.* [19] presented a healthcare blockchain called *FHIRChain* to support identification and permission authorization of health records. But, it does not address the semantic interoperability. Similar works are also carried out for feature selection and extraction of patient clinical data based on DL frameworks for EHR data. Weng *et al.* [20] proposed *DeepChain* that provides a collaborative, distributed federated deep learning model training for incentive fairness and scalability in blockchain. However, the transfer learning function for re-trained models is not discussed appropriately. Pham *et al.* [21] employed LSTM cells for modeling time, admission methods, and suitable diagnosis of patients. To tackle the heavy dictionary of 197,100 unique medical abbreviations, authors in [22] utilized word-embedding approaches by pre-training word model *word2vec*. Suo *et al.* [23] proposed a Convolutional Neural Network (CNN) to capture local trained information as triplets and software cross-entropy loss to form patient clusters. Musaed *et al.* [24] used a transfer learning technique based on VGG-16 and Caffee on existing CNN models. Jacobson *et al.* [25] compared utilized stacked AEs and Restricted Boltzmann Machine (RBM) along-with the *word2vec*-based embedding approach for clinical notes. They found better results compared to [22].

A. Motivation

The aforementioned discussions addresses the issues of deploying EHR in healthcare *4.0* applications. Earlier, authors have also discussed the security issues of deploying EHR in blockchain and proposed solutions like privacy-preserving [15], authentication [16], and authorization using lattices [11]. They did not address the scalability issue of mined transactions. Similarly, other authors proposed self-learning based models based on feature parameters [25], pre-trained models [20], and vector similarity [23]. In healthcare *4.0* applications, security and predictions models forms a close interplay with each other. Hence, motivated from these, this paper proposes a novel framework that integrates lattices, blockchain and DL techniques for healthcare *4.0* applications. The purpose of this paper is two-fold. Firstly, privacy and security of EHR is managed in blockchain by using lattice cryptography, which resists quantum attacks. Blockchain provides immutability and chronology in mined transactions. Secondly, the P-QBN uses DL prediction models on stored featured EHR records. It reduces feature extraction time and improves the scalability of mined transactions by automating recommendations. Also, recommendations are based on past diagnosis to reflect current health conditions. It ensures timely cure and medication for patient by avoiding critical illness which results in saving one's life.

B. Contributions

In this paper, we propose, *BinDaaS*, a framework integrating quantum-resistant blockchain and deep-learning for Healthcare 4.0 environment to secure patient EHR records. Following are the main contributions of the paper-

- An integration of blockchain-deep learning as-a-service to store patient's EHR data in a private and secured manner is proposed.
- Then, a lattice based signature scheme is used to ensure privacy and authentication of EHR records of patients.
- The performance of the proposed framework is validated using various security parameters and predictions models with existing state-of-the art proposals.

C. Organizations

Rest of the paper is organized as follows. Section II describes the system model and problem formulation. The proposed

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

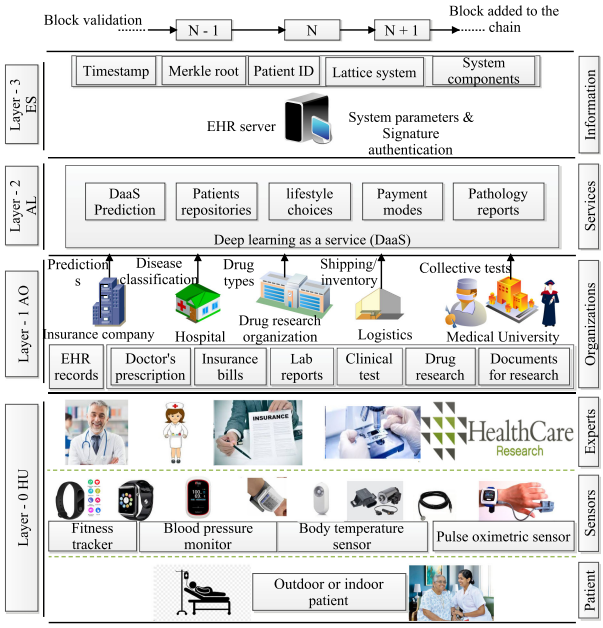


Fig. 1. System architecture of *BinDaaS*.

lattice signature scheme to ensure user privacy and confidentiality in *BinDaaS* is illustrated in Section III. Section IV discusses the integration of DL in *BinDaaS* framework. Section V describes the performance evaluation of the proposed *BinDaaS* framework. Finally, Section VI concludes the paper.

II. System Model and Problem Formulation

This section describes the system model and the problem formulation.

A. System Model

Consider a blockchain based EHR system as shown in Fig. 1, which consists of four layers numbered from *Layer 0* to *Layer 3*. The movement of data is from lowest layer *Layer 0* towards highest layer *Layer 3*. In the system, *Layer 0*, denoted as Healthcare Users (*HU*) comprises of users in the healthcare network, namely-doctors, patients, insurance agents, lab technicians, pathologists, and healthcare researchers. They generate data either in automated mode from automated sensors like-blood glucose biosensors, electrochemical biosensors, amperometic biosensors or record manual data from doctor prescriptions, lab reports, clinical tests, drug research, insurance bills and claims. The collected raw data goes through data level sensor fusion algorithms [26] to achieve homogeneous readings. Important features are classified using bayesian classifiers. Above that, *Layer1*, is the Authoritative Organizational (*AO*) layer. *AO* consists of authorities and stakeholders like- hospitals, medical labs, research organizations, insurance companies, and pharmaceutical companies. They gather the data from *HU* and process them by selecting appropriate feature sets and entity assessment to provide correct treatment to patients. The data is forwarded to *AL* layer through server hypervisors operating over

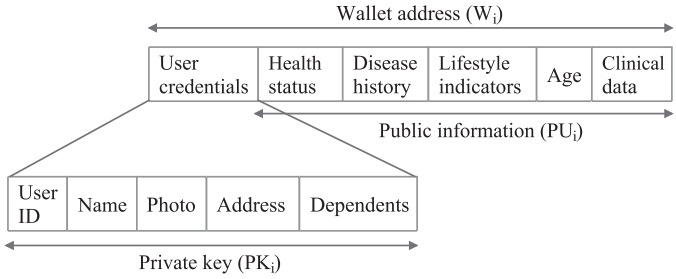


Fig. 2. A patient EHR record structure.

distributed network communication infrastructures like Zwave LTE, Wi-Fi, and Zigbee. Above *Layer 2* is the Analytics Layer (*AL*) which collects data stored at the *AO* layer present in heterogeneous form at various remote locations. *AO* provides decision analytics by invoking proposed *DaaS* algorithm and presents the unified view to *Layer 3*, known as EHR Servers (*ES*). They are required to collect data from *AL* and provide system and operational parameters to secure EHR data. The flow of security parameters is in reverse direction, from *ES* layer towards *HU* layer. The collected parameters ensure user privacy and confidentiality based on lattice signatures and notarization procedures in proposed heterogeneous collective authority authentication mechanism (*HCAAM*) scheme depicted in subsequent sections. Only valid signature blocks are added and mined in the blockchain.

B. Problem Formulation

Authentication and privacy are the two important parameters for EHR records. A consortium allows only medical stakeholders to participate in adding transactions. Patients themselves record or upload prescriptions obtained from doctors in their self EHR records in the chain.

1) The Proposed EHR Record Structure: Consider n users (patients) in the consortium blockchain B denoted by $P_1, P_2, \ldots, P_n$. Once a patient block is mined and added, any i^{th} patient can access the block using his wallet address W_i. Wallet address consists of patient public information PU_i and private credentials PK_i and associated fields as depicted in Fig. 2. Any i^{th} user can access his EHR record by authenticating through a *signed_key*, concatenating both the private key of patient and doctor. The authentication requirements consists of three cases-

1) *CASE-I*: The data collected from *HU* needs to be authenticated at *AO* layer, *for example*, the identity of a doctor needs to be authenticated by a hospital.
2) *CASE-II*: The data at *AO* layer needs to be authenticated by *ES* server.
3) *CASE-III:* Once the authentication parameters are satisfied at the *ES*, a notarization procedure *ES_Notary* is initiated by the ES and subsequently block is added to all users in the network.

2) The Proposed HCAMM Scheme: Once the EHR is stored and recorded, the verification and authorization of EHR by appropriate is required. To facilitate the same, the paper

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

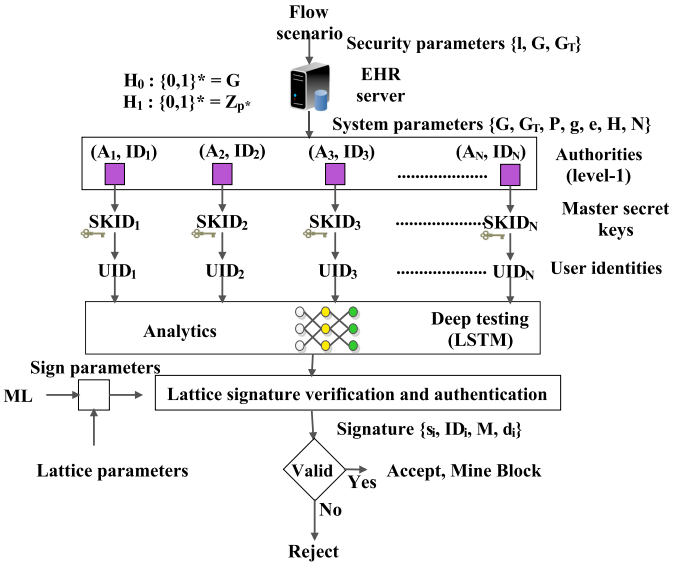


Fig. 3. Proposed *HCAAM* scheme using lattices in *BinDaaS*.

proposes a scheme heterogeneous collective authority authentication mechanism (*HCAAM*) as shown in Fig. 3. The proposed scheme uses lattice based signature generation and verifying operations to ensure privacy and secure sharing of EHR among the various layers of the proposed system model. The recorded transactions in blockchain ensures immutability, and correctness of verifying server parameters. without any intermediaries. The *HCAMM* scheme consists of five sub-schemes, as depicted below-

1) *ES_Param*: The invoking parameters for the whole system is generated by the *ES* which can be public to all other layers. *ES_Param* considers the three-order tuple as follows-

$$ES\_param = (H_s^{512}, \omega, k) \tag{1}$$

where, H_s^{512} represents the signed hash value generated for input data using double SHA-512, ω is a random number generated by any pseudo-random sequence algorithm *like* linear congruential generator, and *k* is the parameter for the discrete log problem.

2) *AO_Param*: It is executed by authorities present in *AO* layer using input parameters *ES_Param* with identities $A_{ID_1}, A_{ID_2}, \ldots, A_{ID_n} \quad i \in (1, N)$. Thus, *AO_Param* considers the following two order tuple,

$$AO\_param = (ES\_Param, A_{ID_i}) \tag{2}$$

where, the i^{th} user is A_{ID_i}. *AO_Param* generates master keys $M_{Ks} \in M_{K_1}, M_{K_2}, \ldots, M_{K_N}$ which are required as the input to the *AO_KeyGen* phase.

3) *AO_KeyGen*: It takes the master keys, the public parameters *ES_Param*, and the user identities $U_{Ks} \in id_1, id_2, \ldots, id_n$ as input to generate the secret keys for the *AO* layer. Thus, *AO_KeyGen* can be expressed as follows,

$$AO\_KeyGen = (M_{Ks}, U_{Ks}, ES\_Param, \delta) \tag{3}$$

where, δ is the random number generator at *AO* layer.

4) *AO_Sign*: It is executed by an authority at the *AO* layer with parameters *AO_KeyGen*, an EHR record *M*, and a digital identity φ_i as follows.

$$\varphi_i = (H_s^{512}(M)||AO\_Keygen) \tag{4}$$

$$AO\_Sign = (\varphi_i, \delta) \tag{5}$$

where, δ ensures diffusion during signing procedure.

5) *HU_Sign*: The heterogeneous data obtained from various sensors or manual records are signed by users like-doctors, labs, insurance agents by using their sign identity id_i, message M, and a secret biometric value $user_{id}$ i.e. the patient personal attributes like fingerprints, and retina-scan. The public key for verifying will be the public parameters from the *ES_Param* algorithm.

6) *ES_Notary*: The notarization algorithm executed by the EHR servers at topmost level takes a message *M*, *AO_Sign* and the *HU_Sign*. If the signatures generated at both the layer matches, *ES* will use *ES_Notary* and sign EHR using it's own key and output *Accept*, else it will output *Reject*. This, notarized document is created as a transaction *T* in the blockchain. The steps are performed for every message *M* in the network. The transactions are now validated using a particular set of nodes, authorized by the EHR servers, to act as *Validators* nodes in the blockchain *B*. *Validators* adds the blocks $B_1, B_2, \ldots, B_{N-1}, B_N$ to the longest valid chain as notarized transactions.

3) Cryptanalysis of Proposed HCAAM Scheme: The following scenario correctly depicts the cryptanalysis performed by an adversary *A* on the *HCAAM* scheme- Suppose a challenger *C* chooses the security parameters *ES_Param* and unknowingly provides the seed values to adversary *A*. Adversary *A*, knowing the seed value, performs a dictionary attack to guess *k*. *A* now presents a list of malicious authorities called *AO** with forged identities as $M_{ks}^* \in ID_1^*, ID_2^*, \ldots, ID_N^*, \quad i \in (1, N)$ to the challenger *C*. The *AO** knowing the *ES_Param* can generate identical duplicate master key sequences $M_{Ks^*} = M_{K_1^*}, M_{K_2^*}, \ldots, M_{K_N^*}$ by invoking *AO_Param* procedure. *A* fools the challenger *C* by following the steps listed below-

1) Adversary issues a request to challenger *C*, $A_{ID_i}, i \subset (1, N)$ to provide master secret key for some genuine authorities. The challenger believes adversary *A* due to previous communication key parameters and provides M_{ks} to be stored in dictionary as (key,value) pairs against stored M_{ks}^*. Note that all dictionary entries need not be filled as only few genuine authority keys are required i.e. $ID_i \in M_{ks}$ to be matched against $ID_i \in M_{ks}^*$.

2) The adversary upon receiving secret key value M_{ks}, looks for *AO_KeyGen* values to compute secret key U_{ks}. Apart from δ, other parameters in the *AO_KeyGen* procedure are public. To compute δ for current communication, *A* launches a traffic analysis attack on the

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

channel to gather the random nonce. Once δ is known, he can solve the discrete logarithm to find U_{ks}. Solving discrete log is a computationally hard problem [27], [28], hence, he can even ask the challenger *C* to communicate with genuine authorities to provide user keys forging as a genuine request. Let U_k be the set of user keys provided by the challenger *C* to *A*.

3) On receiving the user identities, the adversary produces the digital signature φ_i by first computing the AO_sign. As the same nonce δ from *AO_Keygen* phase is used, given a message M, the challenger can now solve $H_s^{512}(M)$, and concatenate it with known sequences from the dictionary by computing the value against the corresponding key in the *AO_KeyGen* procedure.
4) The users at the *HU* layer are made to believe the sign is from trusted source, and they execute the *HU_Sign* procedure. This makes the challenger *C* execute the *ES_Notary* procedure seeing the genuine *HU_Sign*, and forged blocks are now added to the legitimate chain. Thus, the adversary gains illegitimate access to the EHR records of patients.

The adversary A breaks the $HCAMM$ scheme when the following conditions are met-

CASE I: Adversary is able to perform the digital signature φ_i. Thus, the result of *Bitwise-XOR* is zero, since both valid and forged sign matches.

$$\underbrace{(id_i^*, M, user_{id}^*;)}_{Forged-Signature} \bigoplus \underbrace{(id_i, M, user_{id})}_{Original-Signature} = 0 \tag{6}$$

CASE II: Adversary is able to make the users at the *HU* layer believe sign is from trusted source which makes the challenger *C* execute the notarization procedure.

$$\underbrace{ES_{Notary}(id_i^*, M, user_{id}^*;)}_{EHRValidation} = ACCEPT \bigwedge \underbrace{ID^* \notin ID_i}_{(Blockmined)} \tag{7}$$

The above discussion indicates the fact that signing and verifying are the critical operations to ensure the privacy and confidentiality of the stored EHR records. The most conventional cryptographic protocols in the blockchain network are now-a-days susceptible to the quantum attacks [11], [29], since the inception of quantum computers. A quantum computer works on quantum bits, or *qbits*, that can exist in superposition states between 0 ans 1 simultaneously, thus can process information faster than conventional cryptosystems. To thrawt such attacks, quantum resistant signature schemes were proposed [30] based on lattice cryptography.

III. R-LWE: Blockchain Based Lattice Signature Generation and Verification

As mentioned in section II, an adversary *C* can launch signing based attacks over a insecure channel by forging signature patterns and make *HU* layer users accept the forged sign. Thus, to thwart such schemes, the paper proposes a lattice based signature scheme L_S on a blockchain network *B* where for each user U_i, $i \in (1, n)$, we have public/private pairs denoted by PU_i and PK_i.

A. Motivation Behind Lattice Cryptography

Conventional cryptosystems are implemented primarily over finite fields. Symmetric cryptosystems like Data Encryption Standard (DES), Advance Encryption Standard (AES) are now vulnerable to quantum attacks. AES-256, or higher, are still found to be resistant against quantum attacks [31]. In a similar manner, public-key-cryptography (PKC) schemes like Rivest, Shamir, and Adleman (RSA), based on prime integer factorization, Diffie-Hellman (DH), based on secure exchange, and Elliptic Curve Cryptography (ECC), based on computing discrete logarithm problem, are practically broken by quantum computers by employing correct choice of *qbits*.

B. R-LWE: The Proposed Scheme

In the proposed scheme, the EHR server will generate signature parameters based on a lattice scheme over ring learning with errors (R-LWE). It has the mentioned advantages-security against side channel attacks, resist quantum forgeries and collusion attacks, and smaller key sizes. It is also proven to be secure under the random oracle model. A lattice can be defined over ring and fields structure of abstract algebra denoted on a set S with two operations $+$ and $*$. It can be written as $(S, +, *)$. Lattice is defined as arrangements of discrete vectors over a space. Mathematically, it is denoted as follows-

Theorem 1: Let $L = [L_1, L_2, \ldots, L_n] \in R^{p\times p}$ is a $p \times p$ matrix having columns as linearly independent vectors. Now, lattice $L_S \in R^{p\times p}$ is defined as [11], [29]-

$$L_S = \{Lx : x \in \mathbb{Z}^m\} \tag{8}$$

For given prime p, matrix $M \in \mathbb{Z}_q^{n\times m}$ and $u \in \mathbb{Z}_q^n$, a two dimensional q-ary lattice is as follows-

$$L_q(B) \vdash \{f \in \mathbb{Z}^m | Bf \cong 0 \bmod q\} \tag{9}$$

$$L_q^u(B) \vdash \{f \in \mathbb{Z}^m | Bf \cong u \bmod q\} \tag{10}$$

where $L_q(B) = q\dot{L}_q^u(B)^*$ and $L_q^u(B) = q\dot{L}_q(B)^*$ having properties of duals of one another.

Mathematically, a ring reduction is defined as follows [32]-

$$R_{y,[B]} = f = \sum_{i=0}^{n-1} f_i x^i, f_i \in [-B, B] \tag{11}$$

The Gaussian distribution [32] for $d \in \mathbb{Z}$ has a standard deviation σ which is defined with the following parameters

$$L_\sigma = \frac{p_\sigma(C)}{p_\sigma(\mathbb{Z}))}, \sigma > 0,$$
$$p_\sigma(C) = e^{\frac{-c^2}{2\sigma^2}},$$
$$p_\sigma(\mathbb{Z}) = 1 + 2\sum_{c=1}^{\infty} p_\sigma(C) \tag{12}$$

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

EHR SERVER (ES)	*AUTHORITATIVE ORGANIZATIONAL USERS (AO)*
INITIALIZATION PHASE **Inputs**: Security parameters $\{\lambda, G, G_T\}$, master secret key M_{ks}, user identities U_{id_i}, and *PreSeed* μ **Lattice parameters** : $L_{ES}(\omega, \beta_h, \beta_s, n, \sigma, \varrho, q_k)$ **Pseudo-random Function**: $A_1 : (0,1)^{\varrho} \rightarrow (0,1)^{\varrho, \varrho+3}$ **Hash**: $G : (0,1)^* \rightarrow (0,1)^{512}$ **Pseudo-random Function**: $A_2 : (0,1)^* \times (0,1)^{\varrho} \times (0,1)^{512} \rightarrow (0,1)^{\varrho}$. $\xrightarrow{\langle L_{ES}, A_1, A_2, G\rangle}$ **(secure channel) (SSL/TLS)**	**Inputs**: Polynomial $P \in P_1, P_2, \ldots, P_n$ over ring $R_n = (Z)_n[\mathbb{Q}]/[\mathbb{Q}^l + 1]$ Gauss distribution $\sigma((\mathbb{Z})) : (0,1)^{\varrho} \times \mathbb{Z} \rightarrow R$ **Seed Generation**: Using $\sigma(\mathbb{Z})$ for n *HU* users, namely $(seed_1, seed_2, \ldots, seed_n)$ **Encoding**: $E[P]$ defined as two arrays *pos_list* and *sum_list* *pos_list*: $\in [0, 1, \ldots, (n-1)]^G$ *sign_list*: $\in [-1, 1]^{G_T}$ **Sampling Function**: $(0,1)^{\varrho} \times \mathbb{Z} \rightarrow R_{y,[B]}$ **Hash H**: $R_k^q \leftarrow (0,1)^* \times (0,1)^{\varrho}$ polynomials $(p_1, p_2, \ldots, p_k) \in R_q$ compute $[V_1]^M, [V_2]^M, \ldots, [V_k]^M$ for message M $\xleftarrow{\langle P_{R_N}, \sigma_z, E[P], R_{y,[B]}\rangle}$ **(secure channel) (SSL/TLS)**
KEY GENERATION **Inputs**: $P_{RN}, \sigma_Z, E[P], R_{y,[B]}$ Choose a secret value s sampled over σ_Z using the seed from $E[P] \leftarrow e_1, e_2, \ldots, e_n$ **Check**: $\lvert s\rvert < R_N, \lvert E\rvert < L_{ES}$ $counter \leftarrow 1$ For i in σ_z counter $\leftarrow counter + 1$ **Generate Key**: $K_\sigma \leftarrow (K_{\sigma_1}, K_{\sigma_2}, \ldots, K_{\sigma_n})$ **Secret Key**: $S \leftarrow (K_\sigma, \sigma_{\mathbb{Z}})$ Polynomial $P_{red} \leftarrow a_i \sum_{i=1}^{n} p_i + e_i$ $SK \leftarrow$(P$_{red}$, e, K_σ,*nonce*) $\xrightarrow{\langle SK, ID_{ES}, ID_{AO}, nonce\rangle}$ **(secure channel) (SSL/TLS)**	
	SIGNATURE GENERATION **Input**: Message M, SK, random number α, nonce $\chi_1 \leftarrow (SK, \alpha, H(M))$ counter' $\leftarrow counter + 1$ $N \leftarrow$ counter' $\chi_2 \leftarrow (\chi_1, K_\sigma)$ *pos_list* $\in$[0,1,. . .,(n-1)]$^{\chi_1}$ *sign_list* $\in$[-1,1]$^{\chi_2}$ **Signature**: $\mathcal{S} \leftarrow \chi_1 \sigma_{\mathbb{Z}} + \chi_2 SK$ $\xleftarrow{\langle S, SK, counter'\rangle}$ **(insecure channel) public**
SIGNATURE VERIFICATION **Input** :S, $\sigma_{\mathbb{Z}}, M$, nonce, counter' **Choose**: random value r$'_t$ **Check**: $\lvert N\rvert <$counter', accept sign, else discard S for each message M, compute $\chi'_1 \leftarrow (H(M), r', K)$ counter' $\leftarrow counter' + 1$ $C \leftarrow pos\_list, sign\_list$ for each received sign $\mathcal{S}$, compute $\chi'_2 \leftarrow C' - \chi_1 \sigma_2$ if $\chi'_2 = \chi_1 SK^{-1}$ $\chi'_1 = \chi_2 SK^{-1}$ *accept* sign, else *reject*.(**signature matched**)	

Fig. 4. Lattice-based Authentication and Signature verification scheme.

In R-LWE, polynomial multiplications over general field are required. The polynomial p_1, p_2, p_n are defined and let ψ be the primitive n^{th} root of unity in $\mathbb{Z}_q$ with $\psi^n \cong 1 \bmod q$, with δ as primitive $2n^{th}$ root of unity, and $\psi^2 = \delta$. Then, a polynomial is converted in ring-LWE domain by the following formula [33]-

$$\frac{\mathbb{Z}_q[x]}{x^n+1} \rightarrow \mathbb{Z}_q^n, a \Rightarrow \bar{a} = \sum_{i=0}^{n-1} \left(\sum_{j=0}^{n-1} a_j \psi^j \delta_{ij} \right) x^i,$$
$$\mathbb{Z}_q^n \rightarrow \frac{\mathbb{Z}_q[x]}{x^n+1}, \bar{a} \Rightarrow a = \sum_{i=0}^{n-1} \left(n^{-1} \psi^{-i} \sum_{j=0}^{n-1} \bar{a_j} \delta^{-ij} \right) x^i \quad (13)$$

The R-LWE scheme is divided into four phases- initialization, key generation, signature generation, and signature verification. The scheme is depicted in Fig. 4. The phases are as follows-

- *Initialization*: The *ES* layer generates the lattice parameters $L_{ES} = (\omega, \beta_h, \beta_s, n, \sigma, \varrho, q_k)$. The detailed nomenclature of the symbols used in this paper are shown in Table I. Based on L_{ES}, two pseudo-random functions $A_1 : (0,1)^{\varrho} \rightarrow (0,1)^{\varrho, \varrho+3}$ and $A_2 : (0,1)^* \times (0,1)^{\varrho} \times (0,1)^{512} \rightarrow (0,1)^{\varrho}$ is defined. These functions takes as input a *PreSeed* value μ ϱ bits long and maps it to $(\varrho + 3)$ seeds each of ϱ bits. A strong collision hash function G maps a message or record M to constant 512 bits length output. This step is performed so that key generation and signature verification is done on a constant length string. The parameters (L_{ES}, A_1, A_2, G) are now sent by *ES* layer to *AO* layer using a secure channel. The *AO* layer takes as input a polynomial *P* defined over ring R_n employing the Gaussian distribution as

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

TABLE I
ABBREVIATIONS

Symbol	Notations Used
ω	Random Sequence generated by EHR Server at *ES* layer
β_h	# of hash queries
β_s	# of sign queries
A_1	Pseudo-random function 1 at initialization at *ES*
A_2	Pseudo-random function 2 at initialization at *ES*
n	Lattice dimension of (n-1) degree polynomial
σ	Gaussian Distribution
ς	Random nonce for Lattice parameters generation
ϱ	# of RWE samples
$E[P]$	Encoding Function defined on R-LWE
R_k^q	Fixed hash generated at *AO* based on k-seed values
$R_{y,[B]}$	Ring reduction polynomial defined in eq. [11]
P_{R_N}	Polynomial encoding over ring reduction for n users
K_σ	Secret gaussian key distribution
σ_Z	Gaussian distribution over ring (Z)
pos_list	Array depicting position indexes of polynomials defined in encoding list
sign_list	Array of corresponding sign entries at position indexed defined by *pos_list*
SK	Secret-key generated based on lattice parameters
P_{red}	Reduced polynomial as defined in eq. [13]
ID_{ES}	Identity (*IP,MAC*) for EHR Server
ID_{AO}	Identity (*IP,MAC*) for *AO* layer
R_N	Lattice ring for n users
χ_1	Signature generation parameter 1
χ_2	Signature generation parameter 2
α	Random number generation parameter during sign generation phase
χ_1'	Signature verification parameter 1
χ_2'	Signature verification parameter 2
r'	Corresponding random number during matching phase
L_E	Lattice-based encryption

defined in eq. (12). The Gauss sampler generates k-bit seed values for n healthcare users, denoted as $(seed_1, seed_2, \ldots, seed_n)$ based on a nonce counter. An encoding function generates two array lists *pos_list* and *sign_list* with mapping defined as: $E[P] : (0,1)^\varrho \rightarrow (0,1,\ldots,(n-1))^G \times (-1,1)^{G_T}$. Now, the *AO* layer generates fixed hash $H : R_k^q \leftarrow (0,1)^* \times (0,1)^\varrho$, which maps n lattice polynomials to LWE reduction. Lastly, $[V_1]^M, [V_2]^M, \ldots, [V_k]^M$ for any message M is computed for k reduced polynomials out of n polynomials. Now, $(P_{R_N}, \sigma_z, E[P], R_y, [B])$ is sent over secure channel. The authorized users can now login to the system for next key-generation phase.

- *Key Generation*: A secret s is sampled over Gaussian distribution σ_z using the seed from sent $E[P]$ value. Since $E[P]$ contains k reduced polynomials, the seeds are defined as $(seed_s, seed_{e_1}, seed_{e_2}, \ldots, seed_{e_k})$. The checking criteria is that seed value satisfies $|s| < R_N$, i.e. in the range of lattice polynomials. A counter c is initialized to 1 which is incremented with every session key generated to maintain the uniqueness. Secret key S is based on (K_σ, σ_z). Now SK for n users is generated based on reduced polynomial P_{red}. Finally, the secret key parameters $(SK, ID_{ES}, ID_{AO}, nonce)$ is sent over secure channel for n healthcare users to communicate.
- *Signature Generation*: For signing a message *M*, a polynomial p is chosen from R_N and signature generation parameters χ_1 and χ_2 are computed based on SK received from key-generation phase and random number α. The counter is incremented after each generated signature taking values from *pos_list* and *sign_list*. The signature is rejected if $\mathbb{Z} \notin R_{y,[B-L_{ES}]}$. The computed signature $(S, SK, counter')$ is now sent for verification by other party.
- *Signature Verification*: Upon receipt of $\chi_1' \leftarrow (H(M), r,' K)$ for every message M and χ_2 is computed based on selecting values from arrays *pos_list* and *sign_list*. For each received sign $\chi_2 \leftarrow C - \chi_1 \sigma_z$ is computed. It is then matched with original sent values as depicted in figure to accept or reject the sign. It is understood the signature matches for any bit string b', if $2^{(b-1)} - L_E < \lfloor y/2 \rfloor - R_n$.

Algorithm 1. *R-LWE* lattice key generation for secure exchange

Input: *ES* server security parameters (λ, G, G_T), M_{ks}, user id U_{id_i} and *PreSeed* value μ.
Output: Secret Key generation SK and public parameters based on seed value $e \in (e_1, e_2, \ldots, e_n)$.
Initialization: Lattice parameters $L_{ES} = (\omega, \beta_h, \beta_s, n, \sigma, \varrho, q_k)$, pseudo-random functions A_1 and A_2, hash G, and counter=1

1: **for** $i \leftarrow 1$ to n **do**
2: $R_n \leftarrow (Z)_n[\mathbb{Q}]/[\mathbb{Q}^l + 1]$
3: $S \leftarrow (seed_1, seed_2, \ldots, seed_n)$
4: **end for**
5: *PreSeed*$\mu \leftarrow [0,1]^\varrho$
6: *pos_list* $\leftarrow [0, 1, \ldots, (n-1)]^G$
7: *sign_list* $\leftarrow [-1, 1]^{G_T}$
8: **if** $(\sum_{i=1}^n max_i(\mu) > L_{ES})$ **then**
9: $\sigma_z \leftarrow R_{y,[B]}$
10: **else**
11: print *"Reduced Key parameters not generated"* and **return**
12: **end if**
13: **while** ($|S| < R_N$ and $|E| < L_{ES}$) **do**
14: $\sigma_Z \leftarrow (e, counter)$
15: $counter \leftarrow counter + 1$
16: $K_\sigma \leftarrow (K_{\sigma_1}, K_{\sigma_2}, \ldots, K_{\sigma_n})$
17: $S \leftarrow (K_\sigma, seed_a)$
18: $P_{red} \leftarrow (a_i \sum_{i=1}^n P_i + e_i)$
19: $SK \leftarrow (S, e, K_\sigma)$
20: **if** ($|K| <$ *nonce*) **then**
21: print "Session successful, key generated" and **return**
22: **else**
23: print "Session unsuccessful, key not generated" and **return**
24: **end if**
25: **end while**

The proposed algorithm 1 takes as inputs *ES* security parameters, master keys generated at *AO* layer, user identities at *HU* layer, and a *PreSeed* value. Initially, the counter value is set to 1. Lines 1-4 of the algorithm generates the ring polynomial R_N and seed parameters $(seed_1, seed_2, \ldots, seed_n)$, stored in S. Lines 5-8 initializes the arrays *pos_list* and *sign_list*. A security check, called *Reject samples* is now checked in lines 8-12 if they exceed the lattice parametric range. Only those samples which passes the check and in reduction range are allowed to generate secret key based on $\sigma_\mathbb{Z}$ gaussian distribution. Now, secret keys for n users based on reduced values of R-LWE samples are generated. This is depicted in lines 13-20. To ensure freshness and liveliness of the generated

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

Algorithm 2. *R-LWE* lattice signature generation and verification by *BinDaaS* users

Input: Message M, $R_{y,[B]}$, K, random function *rand*, nonce N.
Output: 1 if signatures are verified, and 0 otherwise.
Initialization: counter'=1

1: rand $\leftarrow ((0,1)^{\varrho}, nonce)$
2: $j \leftarrow ySampler$(rand,counter')
3: **for** $i \leftarrow 1$ to k **do**
4: $\quad \chi_1 \leftarrow (SK, \alpha, H(M))$
5: $\quad counter' \leftarrow counter + 1$
6: $\quad c' \leftarrow H(m_{seed_1}, m_{seed_2}, \ldots, m_{seed_k})$
7: **end for**
8: $\chi_2 \leftarrow (\chi_1, K_\sigma)$
9: $pos\_list \in [0, 1, 2, \ldots, (n-1)]^{\chi_1}$
10: $sign\_list \in [-1, 1]^{\chi_2}$
11: $\xi \leftarrow pos\_list, sign\_list$
12: $F \leftarrow c' + \xi$
13: **if** $\mathbb{Z} \notin R_{y,[B-L_{ES}]}$ **then**
14: $\quad counter' \leftarrow counter' + 1$
15: $\quad goto$ step 2 and **Repeat**
16: **end if**
17: Signature $S \leftarrow \chi_1\sigma_z + \chi_2 SK$
18: send $(S, SK, counter')$ for verification at receiver
19: **if** $|N| < counter'$ **then**
20: $\quad$ print "Accept received sign for matching"
21: $\quad$ **if** $(\mathbb{Z} \notin R_{y,[B-L_{ES}]})$ **then**
22: $\quad\quad counter' \leftarrow counter' + 1$
23: $\quad\quad \chi_1' \leftarrow (H(M), r,' K)$
24: $\quad\quad C \leftarrow (pos\_list, sign\_list)$
25: $\quad\quad$ **for** each received sign S **do**
26: $\quad\quad\quad \chi_2' \leftarrow C - \chi_1\sigma_z$
27: $\quad\quad\quad \chi' \leftarrow (\chi_1,' \chi_2')$
28: $\quad\quad$ **end for**
29: $\quad$ **end if**
30: $\quad A \leftarrow \chi_1 SK^{-1}$
31: $\quad B \leftarrow \chi_2 SK^{-1}$
32: $\quad$ **if** $(\chi_2' == A$ and $\chi_1' == B)$ **then**
33: $\quad\quad$ print "Signature match and output 1"
34: $\quad$ **else**
35: $\quad\quad$ print "Signature not match and Output 0"
36: $\quad$ **end if**
37: **end if**

keys, they are check against current counter value. If they are less than pre-specified *nonce* counter, session is successful with key exchange $(K_{\sigma_1}, K_{\sigma_2}, \ldots, K_{\sigma_n})$, otherwise session keys are rejected for all samples as depicted in lines 21-25.

The proposed algorithm 2 takes as input message M, $R_{y,[B]}$ and random function *rand* with a nonce counter N. The purpose of the algorithm is to send and verify the sent signatures over R-LWE. Lines 1-2 initialize the random values over sent seed samples. Now, for a reduced set of k users who wish to sign a message M, lines 3-7 generates signature parameters χ_1 based on $(SK, \alpha, H(M))$. The counter is incremented by 1 for each sign by a user based on secret key generation, as in algorithm 1. Thus, $c' \leftarrow H(m_{seed_1}, m_{seed_2}, \ldots, m_{seed_k})$ denotes k sign operations by users. Lines 8-12 initializes the second sign parameter χ_2 along-with updated values of *pos_list* and *sign_list*. The verification algorithm takes as input signature $S = \chi_1\sigma_z + \chi_2 SK$

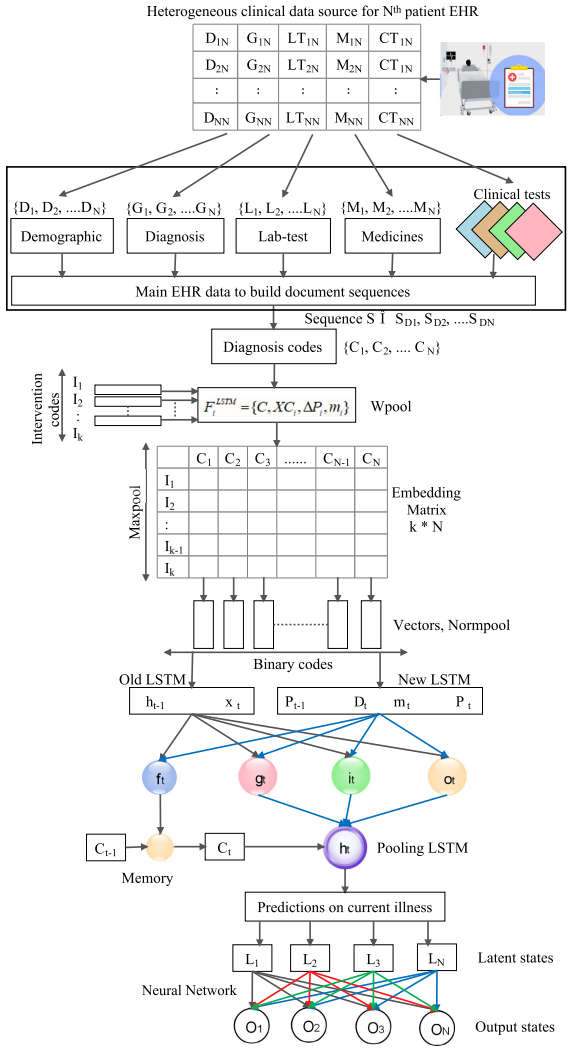


Fig. 5. *DaaS* integration in *BinDaaS* using LSTM for future risk predictions from EHR.

for incremented counter values. A checking criteria, named *Sign Correctness* depicted in line 13-20 ensures every valid signature generated by signing algorithm is only verified by verification algorithm. This ensures that spurious generated signs are discarded. Now, verification parameters χ_1' and χ_2' are computed in lines 19-28. Finally, lines 29-37 computes the modular inverse for secret key and concatenates it with sign parameters, denoted by A and B. This computed values need to match with receiver's verification parameters for sign to be valid.

IV. Integrating DaaS: Deep Learning as-a-Service in BinDaaS

The stored EHR records in blockchain consists of symptoms, clinical observations, monitored data, and sequence of admission of patient, as shown in Fig. 5. The patient admission may be planned (proper appointment from doctor), or unplanned (accidents, heart-stroke). Thus, every records consists of rich information normally consisting of admission time of patient, discharge time, lab reports, diagnosis,

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

Algorithm 3. *LSTM DaaS* for future prediction of diseases

Input: Patient EHR records $D_1, D_2, \ldots, D_n$ as sequence of admissions $S = \{S_{D_1}, S_{D_2}, \ldots, S_{D_n}\}$ for n users.
Patient diagnosis codes $C = \{C_1, C_2, \ldots, C_n\}$ as feature vectors $x_{c_i} \in R^m$
Patient interventions $I = \{I_1, I_2, \ldots, I_k\}$ as feature vectors $x_{I_i} \in R^m$, where m is vector dimension length, elapsed time Δt for each i^{th} patient.
Admission codes from *WPool* with associated probabilities P(*WPool*)
Output: Future prediction of patient health based on outcome probability $P(y|h_{1,2,\ldots,n})$.
Initialization: $i = 0, j = 0$, memory state of LSTM $c = 0$;

for $(i \leftarrow 1$ to $n)$ **do**
  $F^i_{LSTM} \leftarrow \{x_{c_i}, x_{I_i}, \Delta p_i, m_i\}$
  $R^m \leftarrow$ *WPool*$(\varrho_1, \varrho_2, \ldots, \varrho_n)$
  $K \leftarrow$ *Compute_length* $P(y|\varrho_{1,2,\ldots,n})$
  $\Delta P^t_i \leftarrow D^t_i - D^{t-1}_i$
end for
for $(i \leftarrow 1$ to $n)$ **do**
  for $(j \leftarrow 1$ to $k)$ **do**
    $W_{ij} \leftarrow$ *Embed_Matrix* $((D, Z))$
    $B = \{b_0, b_1, \ldots, b_n\}$
    $x^i_t \leftarrow max\{A^{d_1}, A^{d_2}, A^{d_n}\}$
    $p^j_t \leftarrow max\{B^{I_1}_s, B^{I_2}_s, B^{I_k}_s\}$
  end for
end for
for $(i \leftarrow 1$ to $n)$ **do**
  NormPool $\leftarrow m_t + \log(1 + \Delta t)^{-1}$
end for
for $(j \leftarrow 1$ to $k)$ **do**
  WPool $\leftarrow \sigma(\sum_{f=0}^{k-1} w_i x_t + U_t h_{t-1})$
  $A_t \leftarrow \frac{1}{m_t}$(WPool+$b_i$)
  if $(m_t == 1)$ **then**
    $A_t > 0$
  else
    $A_t < 0$
  end if
end for
while (z > 0) **do**
  if $(P > A_t)$ **then**
    $\Delta_{t-1:t} \leftarrow |\log(e + \delta_{t-1:t})^{-1}|$
    $\aleph_i \leftarrow \sigma(w_f x_t + u_f h_{t-1} + Q_f q_{\Delta t-1:t} + p_f p_{t-1} + b_f)$
    SoftMax(z) $\leftarrow e^z / \sum_{z^t} e^{Zt}$
    $P(d_{t+1} = c|f_t) \leftarrow SoftMax(z)$
    MeanPool $\leftarrow h_{1,2,\ldots,n}$
  else
    $MeanPool \leftarrow -\log P(y|u_{1,2,\ldots,n})$
  end if
end while
$e_h \leftarrow \sigma(h_t + b_h)$
$x_y \leftarrow h_t a_n + b_y$
$P(y|h_{1,2,\ldots,n}) \leftarrow f_{prob}(x_y)$

procedures, current lifestyle indicators, medics, and previous illness. Since future illness may depend on long-term dependencies of past records, LSTM model is applicable which inputs a set of discrete elements in time, and embed such sequences as continuous vector sets. The forget gate in LSTM serves for the irregular time gaps between consecutive time steps. The output of the LSTM network is aggregated through a time-decaying pooling strategy for future predictions based on neural nets. Thus, our proposed *DaaS* framework provides end-to-end solution of input EHR records, memorizing a long historical illness through LSTM, and predicting future risks through neural nets.

An admission of any i^{th} patient is a sequence $S = (S_{D_1}, S_{D_2}, \ldots, S_{D_n})$ for EHR record $D = (D_1, D_2, \ldots, D_n)$. Each admission S_{D_i} consists of feature vector set of diagnosis codes $(C_1, C_2, \ldots, C_n)$, which is denoted by $\epsilon_i \in R^m$, where m denotes the vector dimension length. Δt denotes the elapsed time between previous and current admissions respectively. Such time-sequences are recorded for every i^{th} patient, denoted by Δp_i. The LSTM feature vector set for every patient is denoted by-

$$F^i_{LSTM} = \{x_{c_i}, x_{I_i}, \Delta P_i, m_i\} \tag{14}$$

LSTM computes corresponding sequences of distributed illness states $\varrho_1, \varrho_2, \ldots, \varrho_n$ where $\varrho_i \in R^k$, where K is the vector dimension length. The middle layer aggregates these states via multi-state weighted pooling function *WPool*$(\varrho_1, \varrho_2, \ldots, \varrho_n)$ for n scales. The top layer takes such pooled states from *WPool* to compute outcome probability as-

$$P(y|\varrho_{1,2,\ldots,n}) = P(LSTM(WPool)) \tag{15}$$

where $P(y|\varrho_{1,2,\ldots,n})$ depends on outputs and record structure. A record structure can be binary or multiclass in nature. For any patient p, let D from 1 to $|D|$ denotes the set of the variable diagnosis codes and F denotes the set of intervention codes, ranging from 1 to $|F|$. The embedding matrix is of size $n \times k$ with $B \in R^{M \times |F|}$ where A^j_i denotes the i^{th} row element at the j^{th} column for the admission of h diagnosis: $d_1, d_2, \ldots, d_n \in \{1\, to\, |D|\}$. The embedded vectors from matrix are now denoted as $A^{d_1}, A^{d_2}, \ldots, A^{d_n}$ and intervention vectors are $B^{I_1}_s, B^{I_2}_s, \ldots, B^{I_k}_s$. The max pooling *MaxPool* is defined as follows [34]-

$$x^i_t = max\{A^{d_1}, A^{d_2} \ldots, A^{d_n}\} \tag{16}$$

$$p^j_t = max\{B^{I_1}_s, B^{I_2}_s \ldots, B^{I_k}_s\} \tag{17}$$

The normalized sum pooling *NormPool* is defined as [34]-

$$\eta^i_t = \frac{A^{d_1} + A^{d_2} + \ldots + A^{d_n}}{\sqrt{A^{d_1} + A^{d_2} + \ldots + A^{d_n}}}$$
$$\omega^i_t = \frac{B^{I_1}_s + B^{I_2}_s + \ldots + B^{I_k}_s}{\sqrt{B^{I_1}_s + B^{I_2}_s + \ldots + B^{I_k}_s}} \tag{18}$$

The input gate i of the LSTM controls the *NormPool* information to be updated in the memory c. The admission to the LSTM network [34] is defined as-

$$A_t = \frac{1}{m_t}\sigma(w_i x_t + U_t h_{t-1} + b_i)$$
$$m_t = \begin{cases} 1, & \text{admission unit} A_t > 0 \\ 0, & \text{admission unit} A_t < 0 \end{cases} \tag{19}$$

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

The output gate is denoted as o_t and p_0 denotes the intervention weight matrix at time t. The output gate and illness forgetting is moderated by the following equations [34]-

$$o_t = \sigma(w_o x_t + u_0 h_{t-1} + p_o p_t + b_o)$$
$$f_t = \sigma(w_f x_t + u_f h_{t-1} + p_f p_{t-1} + b_f) \quad (20)$$

where p_{t-1} is the intervention embedded vector at time $t-1$, and p_f is the weight matrix of the forget gate. The time decay model of the forget f_t gate is given by

$$f_t \leftarrow d(\Delta_{t-1:t} f_t), \text{where} \Delta_{t-1:t} = |\log(e + \Delta_{t-1:t})^{-1}|$$
$$\aleph_i = \sigma(w_f x_t + u_f h_{t-1} + Q_f q_{\Delta t-1:t} + p_f p_{t-1} + b_f) \quad (21)$$

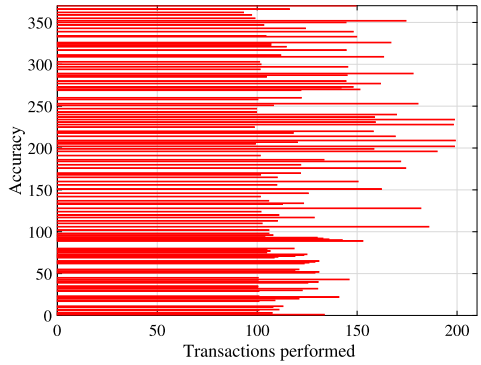
where, $\aleph_i$ denotes the parametric projection in LSTM network. Now, the LSTM units are set up at each discrete time step t with hidden illness state h_t and diagnosis code d_{t+1} for future state is computed using a softmax function $\text{Softmax(z)} = \frac{e^z}{\sum_{Z^t} e^{Z^t}}$. The max-pool chain *HPool* is now defined for n inputs as $h_{1,2,\ldots,n}$ and is given by the equation

$$h_{1,2,\ldots,n} = \frac{1}{s+1} \sum_{o=1}^{n} h_t \quad (22)$$

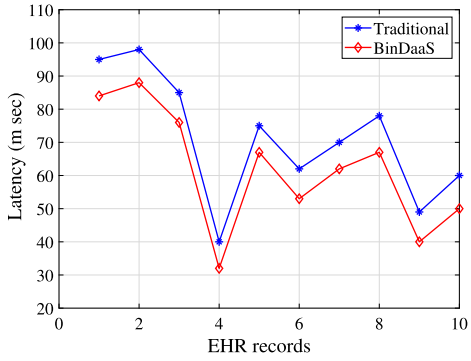
The output of the LSTM network is now fed to the neural network with a single hidden layer. This layer forms predictions for future diseases by forming a stack of the illness predictions as outcome to LSTM. These illness predictions are used to predict future diseases by denoising them using one layer stack autoencoder.

$$e_h = \sigma(h_t + b_h)$$
$$x_y = h_t a_n + b_y$$
$$P(y|h_{1,2,\ldots,n}) = f_{prob}(x_y) \quad (23)$$

The proposed algorithm 3 takes as inputs EHR records, diagnoses codes, and patient interventions as feature vectors. Admission codes based on initial values of *WPool* and the associated probabilities are formed. Initially, the memory state of LSTM network is empty. The algorithm first computes the LSTM feature vector set in lines 1-4 for n users and computes differences in time sequences in line 5. Further, in lines 7-13 embedding matrix of size $n \times k$ is formed based on values of variable diagnoses codes and intervention vectors. The values are normalized in lines 15-16 and stored in *NormPool*. Lines 17-26 denotes the updation of *WPool* based on admission criteria. Lines 28-37 sets up the output of the LSTM network based on *MeanPool* criteria and updates the output probability of illness and risk. Finally, the outputs of LSTM network are fed to simple neural network and activation functions is initiated in Line 38. Line 39-40 performs the computational steps based on single layer stack denoiser which accurately predicts future diseases of patient as probability values which could be used in confusion matrix to generate F-scores.



((a)) Improved accuracy in the *LSTM_DaaS* model



((b)) End-to-end latency over traditional schemes in *BinDaaS*

Fig. 6. Simulation results of *BinDaaS*.

V. Performance Evaluation

In this section, the experimental setup is considered on real-world SemVal 2013 task 9.2 [35] with 27,792 medical EHR training records and 5,716 records in test set. Each EHR has a longitudinal high-dimensional sequence, with ICD-9 clinical events. The selected features are diabetes, obesity, and chronic obstructive pulmonary disease (COPD). The setup is modelled using Python and Keras DL API [36] with TensorFlow [37] backend. The setup is done on NVIDIA GPU of 12 GB. The selection criteria is based on ICD-9 codes for particular disease, and clinical exchanges in 6 months. We split the dataset based on visit date, and updation cycle. We remove overlapping patients suffering from more than one disorder, and remove clinical events that are not updated for more than one year. Finally, the dataset consists of 8,456 patients and 3,140 distinct ICD codes. The selected hyper-parameters uses pre-trained *word2vec* embedding set to 1500, with word vectors updated during training. The hidden-state dimensions are 150. To regularize, we used L1-L2 regularizer, with decay set to 0.003. The number of epochs are 2000 at learning rate of 0.0001. To simulate blockchain setup, we used Corda *v 3.0* [38] and use CordaDApp running Node.js *v 8.8.5* to perform multi-node testing with npm *v 6.7.2*.

A. Simulation Results

The proposed scheme *BinDaaS* is firstly evaluated on the basis of increased accurate predictions using the *DaaS* framework. As indicated in Fig. 6(a), by building the LSTM

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

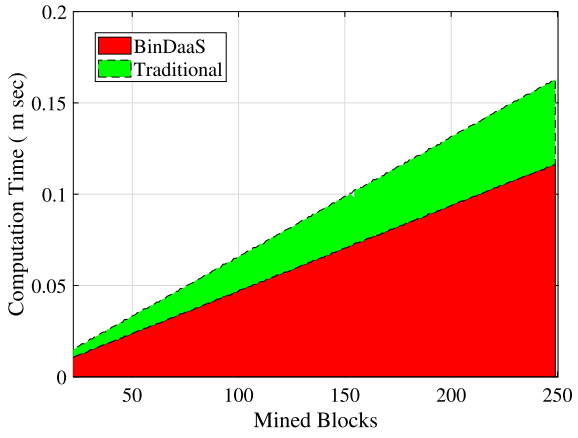


Fig. 7. Comparison of computation time vs. mined blocks of *BinDaaS* over traditional schemes.

network based on patient's EHR records $D_1, D_2, \ldots, D_n$, the proposed algorithm 3 updates the output of the network based on *MeanPool* criteria function, and then a single stack layer denoiser predicts future diseases by confusion matrix. The results obtained shows a considerable improvement in hit rates of True Positives (TP) over False Negatives (FN). The obtained values of proposed *LSTM_DaaS* are TP- 694, TN- 680, FP- 264, and FN- 286. The precision score is 0.7244, recall is 0.7078 and obtained F1-score is 0.7118. Hence, the overall accuracy is increased for prediction of future diseases. Fig. 6(b) indicates the reduced response latency in generating predictions with increasing number of EHR records. The figure clearly states that *BinDaaS* outperforms over other existing state-of-the art schemes. Finally, Fig. 7 indicates the computation time(in msecs) against the number of blocks mined in the overall framework against the traditional approaches. The obtained result indicates the supremacy of the proposed framework over other schemes.

B. Security Evaluation

The computation and communication cost of *BinDaaS* is calculated in this section.

1) Computation Cost: The computation cost is evaluated in four phases, firstly, *initialization phase* is used for generation of key parameters, followed by the *key generation phase* to secure the generation and distribution of secret (symmetric) keys to be used by *HU* users, followed by *signature generation phase* to be used by *AO* users, and lastly *signature verification phase* by *HU* user. As described in [39], the average time required for hash operation is ≈ 0.00032 seconds (sec), Gauss sampling ≈ 0.000265 sec, elliptic curve cryptography (ECC) multiplication and addition takes ≈ 0.0171 sec and 0.0044 sec respectively. Symmetric encryption takes $\approx$ 0.0056 sec, modular exponentiation ≈ 0.0192 sec, modular inverse operation takes ≈ 0.00264 sec, and ring operations $\approx$ 0.0035 sec. Now, cost estimation of each phase is described as follows.

- *Initialization Phase*: The operations used in this phase are one hash function, Gaussian sampling, ECC multiplication, ECC addition, symmetric encryption, two R-LWE, and modular exponentiation operations. Thus, total computation time for initialization phase is $(0.00032 + 0.000265 + 0.0171 + 0.0044 + 0.0056 + 2 \times 0.0035 + 2 \times 0.0192)$sec ≈ 0.073085s or 73.085 milliseconds(ms).
- *Key Generation Phase*: This phase consists of one gauss sampling, and one symmetric encryption. Thus, total computation time for this phase is $(0.000265 + 0.0056)$sec ≈ 0.005865 sec, or 5.865 ms.
- *Signature Generation Phase*: The operations have one R-LWE, hash, and Gauss sampling. Thus, the total computation time for this phase is $(0.0035 + 0.00032 + 0.000265)$ sec ≈ 0.004085 sec or 4.085 ms.
- *Signature Verification Phase*: This phase consists of one Gauss operation, two modular inverse operations, and one hash computation. Thus, total computation time for this phase is $(0.000265 + 0.00032 + 2 \times 0.00264)$ sec ≈ 0.005865 sec or 5.865 ms.

Finally, the overall computation cost for all phases is $(73.085 + 5.865 + 4.085 + 5.865)$ ms ≈ 88.90 ms.

Now, performance analysis of *BINDaaS* has to check for the case when number of transactions are increasing. For this, computation cost for signing and verifying operations is analyzed with reference to the number of transactions and illustrated in Fig. 7. Results shows that signing and verifying periods are linearly varying with the number of transactions, using this the key transfer period of *BINDaaS* is estimated.

2) Communication Cost: Assume user identity to be 128 bits, random nonce 128 bits, and hash identity to be 128 bits in length. The communication cost for various phases are as follows-

- *Initialization Phase*: Initially, the *ES* communicates (L_{ES}, A_1, A_2, G) through a secure channel to *AO*. L_{ES} consists of $(\omega, \beta_h, \beta_s, n, \sigma, \varrho, q_k)$, where ω is random

TABLE II
COMPARISON OF COMPUTATION AND COMMUNICATION COST FOR LATTICE BASED SIGNING AND VERIFYING OPERATIONS

Scheme	Signing Cost	Verifying Cost	Communication Cost	Resisting Collusion Attacks
Guo *et al.* [40]	$(6+t)T_{mul}$ + NT_{pair} + $T_{mtp} \approx 30.614$ ms	$(2tN + 1)T_{pair}$ + $T_{mul} + T_{exp} \approx 301.413$ ms	$(5+t)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_\mathbb{T}\rvert \approx 528$ bytes	✓
Sun *et al.* [41]	$2tT_{mul} + (4t+1)T_{pair}$ + $T_{mtp} + (4t+1)T_{exp} \approx 244.825$ ms	$2tT_{pair}$ + $2tT_{mul} + (3t+1)T_{exp} \approx 150.846$ ms	$\lvert\mathbb{G}\rvert + 2\lvert\mathbb{G}_\mathbb{T}\rvert \approx 280$ bytes	×
Tang *et al.* [42]	$T_{mtp} + 2T_{exp} \approx 12.885$ ms	$3T_{pair}$ + $T_{mtp} + T_{mul} + T_{exp} \approx 24.998$ ms	$2\lvert\mathbb{G}\rvert \approx 80$ bytes	✓
Proposed *BinDaaS*	T_{RLWE} + T_{hash} + $T_{gauss} \approx 4.085$ ms	$2T_{gauss}$ + $2T_{minv}$ + $T_{hash} \approx 5.865$ ms	$5\lvert\mathbb{G}\rvert \approx 212$ bytes	✓

T_{mul}: Modular Multiplication cost; T_{pair}: Bi-linear pairing cost; T_{mtp}: Map-to-map hash cost; T_{exp}: Modular exponentiation cost; T_{RLWE}: Ring learning with errors in $\mathbb{Z}$ over a group $\mathbb{G}$ cost; T_{hash}: Hash operation cost; T_{gauss}: Gaussian sampling cost; T_{minv}: Modular inverse cost; $\mathbb{G}$: Multiplicative cyclic group; $\mathbb{G}_\mathbb{T}$: Another multiplicative cyclic group with $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\mathbb{T}$

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

TABLE III
COMPARISON OF OVERALL COMPUTATION(T_{comp}) AND COMMUNICATION COST(T_{comm}) AGAINST EXISTING SCHEMES

Scheme	T_{comp}	T_{comm}	N_{ex}
Odelu *et al.* [43]	$5T_{eccm} + 2T_{exp} + 12T_{hash} + 2T_{pair} \approx 505.72ms$	240 bytes	3
Wu *et al.* [44]	$10T_{mp} + 2T_{mul} + 5T_{hash} + 2T_{sym} + T_{cer} + T_{cver} \approx 532.43$ ms	456 bytes	4
Kim *et al.* [45]	$30T_{hash} + 3T_{sym} \approx 26.40$ ms	544 bytes	2
Hathaliya *et al.* [46]	$9T_{hash} + 3T_{sym} + 4T_{eccm} \approx 96.64$ ms	176 bytes	3
Aujla *et al.* [47]	$2T_{gauss} + 1T_{FA} + 2T_{FF} + T_{enc} + T_{dc} \approx 5.665$ ms	228 bytes	4
Proposed *BinDaaS*	$3T_{hash} + 4T_{gauss} + T_{eccm} + T_{ecca} + 3T_{RLWE} + T_{sym} + 2T_{minv} \approx 88.90$ ms	212 bytes	4

N_{ex}: # of messages exchanged; T_{eccm}: ECC multiplication cost; T_{ecca}: ECC addition cost; T_{exp}: Modular exponentiation cost; T_{hash}: Hash output cost; T_{pair}: Bi-linear pairing cost; T_{mp}: Multiplication point time cost; T_{mul}: Modular multiplication cost; T_{sym}: Symmetric encryption cost; T_{cer}: Certificate cost; T_{cver}: Certificate verification cost; T_{gauss}: Gauss sampling cost; T_{FA}: Fourier addition cost; T_{FF}: Fourier forward cost; T_{enc}: Encryption cost; T_{dc}: Decryption cost; T_{RLWE}: Ring learning with errors in $\mathbb{Z}$ over a group $\mathbb{G}$ cost; T_{minv}: Modular inverse cost.

nonce of 128 bits, β_h and β_s are hash identities, so $(128 + 128 = 256)$ bits, ϱ is used as mapping to $\varrho + 3$ in functions A_1 and A_2, respectively. The mapping is a hash identity i.e. 128 bits in length, Now, *AO* sends $(P_{R_N}, \sigma_{\mathbb{Z}}, E[P], R_{y,[B])}$ for key generation. This includes just the sampling function using seed value, i.e. 128 bits and hash identity i.e. 128 bits. Thus, total cost for this phase is $(128 + 256 + 128 + 128 + 128) \approx 768$ bits.

- *Key Generation Phase*: This phase sends $(SK, ID_{ES}, ID_{AO}, nonce)$ for signature generation. SK uses identity exchanges of *ES* and *AO* which is bidirectional, i.e. $128 + 128 = 256$ bits, and a random nonce of 128 bits. Thus, communication bits processed in this phase is ≈ 384 bits.
- *Signature Generation Phase*: This phase exchanges $(S, SK, counter')$. Signature S employs hash identity of 128 bits, SK uses 256 bits, and counter value is maximum 32 bits to support 2^{32} operations at maximum. Thus, bits exchanged are $128 + 256 + 32 \approx 416$ bits.
- *Signature Verification Phase*: This phase verifies the identity, hence 128 bits as signature form.

Thus, the total communication cost (CC) of the scheme is $768 + 384 + 416 + 128 \approx 1696$ bits. Table II shows the comparative analysis with existing approaches in terms of computation and communication cost, while Table III shows the overall computation and communication cost against existing state-of-the art cryptographic schemes.

C. Comparative Analysis

The proposed scheme *BinDaaS* is now compared against existing state-of-the art schemes. *BinDaaS* employs lattice cryptography against traditional cryptosystems for signature scheme. All the EHR records are stored in the blockchain network which provides the notion of auditability, transparency and trust among distributed users. Table IV shows the comparative analysis of the proposed scheme against other existing cryptosystems. The results clearly demonstrate that the proposed scheme has higher security against the chosen parameters on the basis of known attacks in cryptosystems.

TABLE IV
COMPARATIVE ANALYSIS WITH EXISTING SCHEMES

Parameters	Bao *et al.* [48]	Li *et al.*[49]	Hathaliya *et al.*[46]	Aujla *et al.* [47]	Proposed *BinDaaS*
A1	✓	×	✓	✓	✓
A2	×	×	✓	✓	✓
A3	×	×	✓	✓	✓
A4	×	×	✓	✓	✓
A5	×	×	×	×	✓
A6	-	✓	✓	×	✓
A7	×	×	×	×	✓
A8	×	-	×	✓	✓
A9	×	×	×	✓	✓
A10	-	×	✓	✓	✓

A1: Replay Attacks; A2: Side-Channel Attacks; A3:Distributed Denial-of-Service (DDoS) attacks; A4: Session-based attacks A5: Provenance and auditability attacks; A6: Tracebility of attacks; A7: Signature-forgery attacks; A8: Signature verificability; A9; Quantum attacks; A10: Known ciphertext attack; ✓ shows scheme is safe; × shows scheme is not safe; & - shows attack is not considered in the scheme.

VI. CONCLUSION

The shift of users and services in the Healthcare 4.0 demands decentralization and at the same time provides necessary requirements of user privacy and confidentiality. Moreover, an appropriate decisions based on previous EHR records of the user needs to be analyzed. The proposed architecture *BinDaaS* provides an integrating framework to ensure security through blockchain and also provides future risk prediction of diseases of patient so that accurate preventions for critical illness can be taken care to save precious life of patients. *BinDaaS* addresses three key contributions- (i) Integrating DL and blockchain to securely store the patient EHR data and provides future predictions based on past repositories, (ii) A lattice-based key and signature verification scheme to resist quantum attacks, and (iii) validation of the security scheme and prediction model against existing state-of-the art infrastructures. Security evaluation is designed on Lattice cryptosystem and is analyzed based on computation and communication cost. The obtained results indicate the supremacy of the proposed framework. The communication cost based on lattice model consists of large number of parameters based on Gaussian distributions. This increases the required cost, which is a critical issue. The above can be addressed in future by reducing parameters generated during initialization phase at the same desired security level. In future, the proposed framework will include the historical EHR of patients that would be trained based on graphical similarity of node degree and diameter.

REFERENCES

[1] N. Kabra, P. Bhattacharya, S. Tanwar, and S. Tyagi, "Mudrachain: Blockchain-based framework for automated cheque clearance in financial institutions," *Future Gener. Comput. Syst.*, vol. 102, pp. 574–587, 2020.

[2] A. Srivastava, P. Bhattacharya, A. Singh, A. Mathur, O. Prakash, and R. Pradhan, "A distributed credit transfer educational framework based on blockchain," in *Proc.* 2nd *Int. Conf. Advances Comput., Control Commun. Technol.*, Allahabad, India, 2018, pp. 54–59.

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

[3] P. Bhattacharya, S. Tanwar, R. Shah, and A. Ladha, "Mobile edge computing-enabled blockchain framework—A survey," in *Proc. Int. Conf. Recent Innov. Comput.*, P. K. Singh, A. K. Kar, Y. Singh, M. H. Kolekar, and S. Tanwar, Eds. Cham, Switzerland: Springer, 2020, pp. 797–809.

[4] U. Bodkhe, P. Bhattacharya, S. Tanwar, S. Tyagi, N. Kumar, and M. Obaidat, "Blohost: Blockchain enabled smart tourism and hospitality management," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst.*, Beijing, China, 2019, pp. 1–5.

[5] I. Mistry, S. Tanwar, S. Tyagi, and N. Kumar, "Blockchain for 5G-enabled IoT for industrial automation: A systematic review, solutions, and challenges," *Mech. Syst. Signal Process.*, vol. 135, pp. 1–21, 2020.

[6] "Healthcare data management meet blockchain," Accessed: Mar. 28, 2019. [Online]. Available: https://steemit.com/healthcare/@robmenzies/healthcare-datamanagement-meet-blockchain

[7] J. Vora *et al.*, "Bheem: A blockchain-based framework for securing electronic health records," in *Proc. IEEE Globecom Workshops*, Abu-Dhabi, UAE, 2018, pp. 1–6.

[8] A. Act, "Health Insurance Portability and Accountability Act of 1996," U.S. Public Law 104–191, U.S. House of Rep., Washington, DC, USA, 1996.

[9] R. Gupta, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, and B. Sadoun, "Habits: Blockchain-based telesurgery framework for healthcare 4.0," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst.*, Aug. 2019, pp. 1–5.

[10] S. Tanwar, K. Parekh, and R. Evans, "Blockchain-based electronic healthcare record system for healthcare 4.0 applications," *J. Inf. Secur. Appl.*, vol. 50, 2020, Art. no. 102407.

[11] C.-Y. Li, X.-B. Chen, Y.-L. Chen, Y.-Y. Hou, and J. Li, "A new lattice-based signature scheme in post-quantum blockchain network," *IEEE Access*, vol. 7, pp. 2026–2033, 2019.

[12] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for AI: Review and open research challenges," *IEEE Access*, vol. 7, pp. 10127–10149, 2019.

[13] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[14] P. M. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[15] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K.-K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Comput.*, vol. 5, no. 1, pp. 31–37, Jan./Feb. 2018.

[16] L. Chen, W.-K. Lee, C.-C. Chang, K.-K. R. Choo, and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing," *Future Gener. Comput. Syst.*, vol. 95, pp. 420–429, 2019.

[17] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, and N. Kumar, "An efficient and practical smart card based anonymity preserving user authentication scheme for TMIS using elliptic curve cryptography," *J. Med. Syst.*, vol. 39, pp. 1–18, Oct. 2015.

[18] P. Gope, R. Amin, S. H. Islam, N. Kumar, and V. K. Bhalla, "Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment," *Future Gener. Comput. Syst.*, vol. 83, pp. 629–637, 2018.

[19] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "FHIRChain: Applying blockchain to securely and scalably share clinical data," *Comput. Struct. Biotechnol. J.*, vol. 16, pp. 267–278, 2018.

[20] J.-S. Weng, J. Weng, M. Li, Y. Zhang, and W. Luo, "DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IACR Cryptol. ePrint Arch.*, vol. 14, pp. 1–18, 2019.

[21] T. Pham, T. Tran, D. Phung, and S. Venkatesh, "DeepCare: A deep dynamic memory model for predictive medicine," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, Macau, China, 2016, pp. 30–41.

[22] Y. Liu, T. Ge, K. Mathews, H. Ji, and D. McGuinness, "Exploiting task-oriented resources to learn word embeddings for clinical abbreviation expansion," in *Proc. BioNLP*, Beijing, China, Jul. 2015, pp. 92–97.

[23] Q. Suo *et al.*, "Deep patient similarity learning for personalized healthcare," *IEEE Trans. Nanobiosci.*, vol. 17, no. 3, pp. 219–227, Jul. 2018.

[24] M. Alhussein and G. Muhammad, "Voice pathology detection using deep learning on mobile healthcare framework," *IEEE Access*, vol. 6, pp. 41034–41041, 2018.

[25] O. Jacobson and H. Dalianis, "Applying deep learning on electronic health records in Swedish to predict healthcare-associated infections," in *Proc. 15th Workshop Biomed. Natural Lang. Process.*, Berlin, Germany, Aug. 2016, pp. 191–195.

[26] R. Gravina, P. Alinia, H. Ghasemzadeh, and G. Fortino, "Multi-sensor fusion in body sensor networks: State-of-the-Art and research challenges," *Inf. Fusion*, vol. 35, pp. 68–80, 2017.

[27] N. Kumar, N. Chilamkurti, and S. C. Misra, "Bayesian coalition game for the internet of things: An ambient intelligence-based evaluation," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 48–55, Jan. 2015.

[28] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Inf. Sci.*, vol. 321, pp. 263–277, 2015.

[29] D. He, S. Zeadally, N. Kumar, and J. Lee, "Anonymous authentication for wireless body area networks with provable security," *IEEE Syst. J.*, vol. 11, no. 4, pp. 2590–2601, Dec. 2017.

[30] W. Yin, Q. Wen, W. Li, H. Zhang, and Z. Jin, "An anti-quantum transaction authentication approach in blockchain," *IEEE Access*, vol. 6, pp. 5393–5401, 2018.

[31] D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen, "Efficient privacy-aware authentication scheme for mobile cloud computing services," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1621–1631, Jun. 2018.

[32] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2010, pp. 1–23.

[33] N. Bindel *et al.*, "Submission to NIST's post-quantum project: Lattice-based digital signature scheme qTESLA," 2018.

[34] T. Pham, T. Tran, D. Phung, and S. Venkatesh, "Predicting healthcare trajectories from medical records: A deep learning approach," *J. Biomed. Informat.*, vol. 69, pp. 218–229, 2017.

[35] B. Xu, X. Shi, Z. Zhao, and W. Zheng, "Leveraging biomedical resources in Bi-LSTM for drug-drug interaction extraction," *IEEE Access*, vol. 6, pp. 33432–33439, 2018.

[36] F. Chollet *et al.*, "Keras: The Python deep learning library," Astrophysics Source Code Library, 2018.

[37] A. Vishnu, C. Siegel, and J. Daily, "Distributed tensorflow with MPI," 2016, *arXiv:1603.02339*.

[38] D. Mohanty, *Corda Architecture*. Berkeley, CA, USA: A Press, 2019, pp. 49–60.

[39] J. Srinivas, A. K. Das, N. Kumar, and J. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 5, pp. 942–956, Sep.–Oct. 2020.

[40] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018.

[41] Y. Sun, R. Zhang, X. Wang, K. Gao, and L. Liu, "A decentralizing attribute-based signature for healthcare blockchain," in *Proc. 27th Int. Conf. Comput. Commun. Netw.*, Hangzhou, China, 2018, pp. 1–9.

[42] F. Tang, S. Ma, Y. Xiang, and C. Lin, "An efficient authentication scheme for blockchain-based electronic health records," *IEEE Access*, vol. 7, pp. 41678–41689, 2019.

[43] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1900–1910, May 2018.

[44] D. Wu and C. Zhou, "Fault-tolerant and scalable key management for smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 375–381, Jun. 2011.

[45] H. J. Kim and H. S. Kim, "AUTH HOTP-HOTP based authentication scheme over home network environment," in *Proc. Int. Conf. Comput. Sci. Its Appl.*, 2011, pp. 622–637.

[46] J. J. Hathaliya, S. Tanwar, S. Tyagi, and N. Kumar, "Securing electronics healthcare records in healthcare 4.0: A biometric-based approach," *Comput. Elect. Eng.*, vol. 76, pp. 398–410, 2019.

[47] G. S. Aujla, R. Chaudhary, K. Kaur, S. Garg, N. Kumar, and R. Ranjan, "SAFE: SDN-assisted framework for edge–cloud interplay in secure healthcare ecosystem," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 469–480, Jan. 2019.

[48] H. Bao and R. Lu, "Comment on 'privacy-enhanced data aggregation scheme against internal attackers in smart grid'," *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 2–5, Feb. 2016.

[49] X. Li, M. H. Ibrahim, S. Kumari, A. K. Sangaiah, V. Gupta, and K.-K. R. Choo, "Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks," *Comput. Netw.*, vol. 129, pp. 429–443, 2017.

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a

Pronaya Bhattacharya is currently working toward the Ph.D. degree in optical networks with Dr. A.P.J. Abdul Kalam Technical University, Lucknow, India. He is an Assistant Professor with Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad, India. He has authored or coauthored more than 15 research papers published in leading journals and conferences of Elsevier, Springer, ACM, and IEEE. His current research interests include optical networks, computational aspects in wireless networks and blockchain technology. He is a Reviewer for the board of international journals—*Journal of Engineering Research*, Kuwait University, *Journal of Optical Communications*, DeGruyter, *Security and Privacy Journal*, Wiley, and *International Journal of Communication Systems*, Wiley. He is a lifetime member of professional organizations like ISTE and IAENG. One of his works has been awarded the Best Research Paper Award in Springer ICRIC-2019.

Sudeep Tanwar received the Ph.D. degree in computer science and engineering with specialization in wireless sensor network from Mewar University, Chittorgarh, India. He is as Associate Professor with Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad, India. He is a Visiting Professor with Jan Wyzykowski University, Polkowice, Poland and the University of Pitesti, Pitesti, Romania to carry out scientific activities. He has authored or coauthored more than 100 technical research papers in leading journals and conferences from the IEEE, Elsevier, Springer, John Wiley, and others. Some of his research findings are published in top-cited journals such as the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON VEHICULAR NETWORKS, IEEE NETWORK MAGAZINE, *Wireless Communication Magazine*, IEEE SYSTEMS JOURNAL, *Computer and Electrical Engineering*, *Journal of Network and Computer Application*, *Future Generation Computer Systems*, *Applied Soft Computing*, *Pervasive and Mobile Computing*, *International Journal of Communication System*, and *Telecommunication Systems*. He has authored or edited six books. He has won academic/research awards at the National level and three best research paper awards, two of which are from top tier conferences of IEEE ComSoc (IEEE ICC and IEEE-GLOBECOM). He is an Associate Editor of IJCS-Wiley, *Security and Privacy International Journal*—Wiley. He has been invited as a Guest Editors or Editorial Board Members of many international journals, invited for keynote Speaker in many international conferences held in Asia and invited as a Program Chair, a Publications Chair, a Publicity Chair, and a Session Chair in many conferences held in North America, Europe, Asia, and Africa. His current research interests include wireless networks, blockchain technology, smart grid, fog computing, healthcare 4.0, and computational aspects of 5G.

Umesh Bodkhe is currently working toward the Ph.D. degree with Nirma University, Ahmedabad, India. He is an Assistant Professor with Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad, India. His current research interests include network security and blockchain technology. He is a lifetime member of ISTE.

Sudhanshu Tyagi (Senior Member, IEEE) received the Bachelor in Engineering degree in electronics and telecommunication from North Maharashtra University, Jalgaon, India, in 2000, the Masters in Technology (with Hons.) degree in electronics and communication engineering from National Institute of Technology, Kurukshetra, India, in 2005, and the Ph.D. degree from Mewar University, Chittorgarh, India, in 2016. He is an Assistant Professor with the Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Deemed University, Patiala, India. He is a Visiting Professor with Jan Wyzykowski University, Polkowice, Poland. He achieved second rank in the university at under graduation level. He has 60 research publications in peer-reviewed journal and conferences from leading publishers like Elsevier, Springer, Wiley, etc. He is a Reviewer for the board of international journal from leading publisher like *International Journal of Ad-Hoc and Ubiquitous Computing*, Inderscience and *Journal of the Franklin Institute*, Elsevier. His research interests include lifetime enhancement of homogeneous and /or heterogeneous WSNs. He is a Senior Member of the IEEE.

Neeraj Kumar (Senior Member, IEEE) received the Ph.D. degree in computer science engineering from Shri Mata Vaishno Devi University, Katra, India. He was a Postdoctoral Research Fellow with Coventry University, Coventry, U.K. He is currently working as a Professor with the Department of Computer Science and Engineering, Thapar University, Patiala, India. He has authored or coauthored more than 400 technical research papers in leading journals and conferences from IEEE, Elsevier, Springer, John Wiley, and others. Some of his research findings are published in top cited journals such as IEEE TIE, IEEE TDSC, IEEE TITS, IEEE TCC, IEEE TKDE, IEEE TVT, IEEE TCE, IEEE Netw., IEEE Comm., IEEE WC, IEEE IoTJ, IEEE SJ, FGCS, JNCA, and ComCom. He has guided many Ph.D. and M.E./M.Tech. students. His research is supported by fundings from Tata Consultancy Service, Council of Scientific and Industrial Research, and Department of Science and Technology. He was the recipient of the Best Research Paper awards from IEEE ICC 2018 and IEEE Systems Journal 2018. He is the highly cited researcher of 2019 as per the WoS list. He is leading the research group Sustainable Practices for Internet of Energy and Security where group members are working on the latest cutting-edge technologies. He is a TPC member and a Reviewer of many international conferences across the globe. He is a Visiting Professor with Coventry University, Coventry, U.K. He is in the editorial board of JNCA, Elsevier, IEEE COMMUNICATION MAGAZINE, IEEE NETWORK MAGAZINE, Elsevier, *Computer Communications*, IJCS, Wiley, and *Security & Privacy*, Wiley. He is an Adjunct Professor with KAU, Saudi Arabia and Asia University, Taiwan. He is a Senior Member of the IEEE.

rized licensed use limited to: AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF ENGINEERING. Downloaded on November 23,2021 at 04:55:31 UTC from IEEE Xplore. Restrictions a