Fortified-Chain 2.0: Intelligent Blockchain for Decentralized Smart Healthcare System

Bhaskara S. Egala, Ashok K. Pradhan, *Member, IEEE*, Prasenjit Dey,
Venkataramana. Badarla, *Senior Member, IEEE*, and Saraju P. Mohanty, *Senior Member, IEEE*

***Abstract*—The Internet of Medical Things (IoMT) technology's fast advancements aided smart healthcare systems to a larger extent. IoMT devices, on the other hand, rely on centralized processing and storage systems because of their limited computational and storage capacity. The reliance is susceptible to a single point of failure (SPoF) and erodes the user control over their medical data. In addition, Cloud models result in communication delays, which slow down the system's overall reaction time. To overcome these issues a decentralized distributed smart healthcare system is proposed that eliminates the SPoF and third-party control over healthcare data. Additionally, the proposed Fortified-Chain 2.0 uses a blockchain-based selective sharing mechanism with a mutual authentication technique to solve the issues such as data privacy, security and trust management in decentralized peer-to-peer healthcare systems. Also, we suggested a hybrid computing paradigm to deal with latency, computational and storage constraints. A novel distributed Machine Learning (ML) module named Random Forest Support Vector Machine (RFSVM) also embedded into the Fortified-Chain 2.0 system to automate patient health monitoring. In the RFSVM module, Random Forest (RF) is used to select an optimal set of features from patients data in real time environment and also Support Vector Machine (SVM) is used to perform the decision making tasks. The proposed Fortified-Chain 2.0 works on a private blockchain-based Distributed Data Storage System (DDSS) that improves the system-level transparency, integrity, and traceability. Fortified-Chain 2.0 outperformed the existing Fortified-Chain in terms of low latency, high throughput, and availability with the help of a mutual authentication method.**

***Index Terms*—Blockchain, Hybrid Computing, Machine Learning, Security and Privacy, Random Forest (RF), Support Vector Machine (SVM), Smart Healthcare System.**

I. Introduction

The widespread availability and application of Internet of Medical Things (IoMT) in smart healthcare services enable patients to provide a real-time treatment with the minimal effort [1, 2]. Moreover, IoMT devices enable the service providers to monitor and alert patient's health status in a real-time environment. In traditional smart healthcare system the IoMT collects, stores, processes, and analyses patient data on a centralized cloud platform. In addition, cloud-centric models depend on third-party data security and privacy mechanisms that lead to minimise the patient control over their medical data. However, the centralized models produce high bandwidth traffic and more latency during communication and computation. The centralized architectures are inappropriate for healthcare applications where low latency, high availability, and transparency are crucial factors. In addition, cyber criminals shifted their focus on IoMT based healthcare applications where the security mechanisms are comparatively weaker because of IoMT constraints [3–5].

Consequently, researchers are focused on decentralised IoMT security and privacy issues to find out the suitable architectural solutions [6] as the centralized models are ambiguous regarding their data management and also susceptible to SPoF. Since, data availability and transparency are significant concerns in future healthcare systems, therefore there is a huge requirement for decentralized models with high data availability and transparency. However, the present cloud-centric models are low in cost and provide industry standards security and privacy for the medical data in spite it is controlled by a third party. The primary security services like confidentiality, integrity, and availability are still doubtful from a cloud-centric model whenever the platform is compromised [7]. Unauthorized modifications are untraceable when it occurs from the service provider's end. As a result, the healthcare domain reconstructs the present healthcare system architectures which is based on a user-centric decentralized model. Though the decentralized models eliminate SPoF, but still they face issues related to data privacy and security. Likewise, decentralized models also have data storage and data management problems because of their distributed nodes. The proposed system with a tamper-proof distributed public ledger eliminates the single node-centric services by providing process sharing and distributing the data storage tasks with all other nodes participating in that system.

In recent times, the latest technologies such as blockchain and edge computing provide an effective solution to the decentralized healthcare applications. The blockchain technology guarantees data integrity, transparency, and traceability in a distributed environment. Whereas, the edge computing paradigm helps the healthcare systems to collect and process the critical data locally [8]. The edge computer minimize the latency during communication and also simplifies the real-time data utilization. The combination of cloud and edge computing paradigms can able to solve many issues related to the latency and computation [9]. Edge computing is in its early stage, where security and privacy standards are yet to be

Bhaskara S. Egala, Dept. of Computer Sci. and Eng., SRM University-A.P, IN, E-mail: bhaskara santhosh@srmap.edu.in

Ashok K. Pradhan (Corresponding Author) Dept. of Computer Sci. and Eng., SRM University-A.P, IN E-mail: ashokkumar.p@srmap.edu.in

Prasenjit Dey Dept. of Computer Sci. and Eng., Cooch Behar Government Eng. College, Cooch Behar, IN prasenjitdey.dey@cgec.org.in

Venkataramana.B Dept. of Computer Sci. and Eng., IIT- Tirupathi, E-mail: ramana@iittp.ac.in

Saraju P. Mohanty Dept. of Computer Sci. and Eng., University of North Texas, USA, E-mail: saraju.mohanty@unt.edu

Manuscript received xx, 202x; revised xx, 202x.

finalized. With the support of cryptography and public ledger systems hybrid computing can overcome these issues. There are few proven decentralized file-sharing technologies such as InterPlanetary File System (IPFS), which support healthcare systems to manage distribution of files and control their different file versions. On the other hand, manually processing and analyzing the huge patient data, which is generated from the healthcare systems in a real time environment is very much difficult. Due to this, currently many research works are focused on embedding machine learning (ML) and data science in healthcare systems. A framework called distributed AI/ML models (D-AI/ML) [10] helps the healthcare sector to migrate from cloud-centric to decentralised intelligence model. The integration of D-AI/ML with edge computing improves the system's real-time response rate with minimum latency. Traditionally, patient medical data is restricted to one hospital or at most to one group of hospitals. The recent healthcare crises underlined the importance of open medical data exchange between diverse stakeholders to improve the medical services[11]. As a result, community-based control models for autonomous healthcare services are required, where all stakeholders share their equal power and accountability.

Rest of this paper is organized as follows. Section II discusses the contribution of this article. Section III illustrates the contemporary of related works. The preliminaries and working model of the proposed architecture is discussed in Section IV. System specific security and privacy mechanisms are introduced in Section V. We compared the proposed system with the existing works, and demonstrates the experiment results and its security analysis in Section VI. Finally, conclusion and future work is discussed in Section VII.

II. Contributions of the Current Paper

A. The Problem Addressed in the Current Paper

The classical centralised server-centric models need to store and process huge data set which is controlled by a third-party. In addition, the centralised systems are prone to SPoF and needs high bandwidths for communication. Though the decentralised system provides better solutions for the above concerned issues, however it is doubtful for the data privacy, security and availability issues. Moreover, latency and accuracy are crucial when a decision is made from decentralised database. The proposed work focuses on a decentralised distributed storage system-based data security, privacy, traceability and transparency when no centralised trust mechanism is available.

B. The Contributions

A novel blockchain-based community-controlled healthcare prototype is proposed to address the medical data security and privacy in a distributed computing environment by establishing trust between diffrent stakeholders. Further, a patient-centric access control named as Selective Sharing Mechanism (SSM) is suggested to access actors data. In order to detect and alert abnormality in heart rate in a real time scenario a novel machine learning algorithm called RFSVM is proposed for the ICU patients. A patient-centric medical service ecosystem is introduced by adopting the technologies such as DDSS, Blockchain, and Hybrid computing respectively.

C. The Challenges in Solving the Problem

Implement a decentralised distributed blockchain-based cryptographical security and privacy mechanism is a challenging task due to IoMT resource-constraints. Establishing trust between participating nodes and actors without a centralised trust management system is one of the known issues for decentralised distributed healthcare systems. Design and integrate a machine learning model on DDSS for real-time assisting with minimal latency is another challenge.

D. Significance of the Contributions

Mutual authentication mechanism helps the devices to initiate a secure communication without interference of edge nodes. Due to this, inter-device communication is faster. The RFSVM helps the system to identify patient's heart related critical information by employing DDSS in a real-time environment. The Off-chain cache helps the system to minimize the latency and increase the system throughput. A simplified Hyperledger-based Fortified-Chain 2.0 is compared with Ethereum-based Fortified-chain to achieve improvements in scalability and performance. The trust establishment between the stakeholders is achieved with smart contracts and public ledger on DDSS.

III. Related Work

In this section, we suggested a comprehensive review of literature related to smart healthcare systems security and privacy issues. A holistic comparison of related works with our proposed system is presented in table.I. The authors in [12] presented IoMT trends where they illustrated a patient monitoring system prototype with the help of Wireless Body Area Networks (WBAN). An extended work of WBAN is demonstrated in [13] which include blockchain and IPFS to overcome transparency and traceability issues. The authors in [14] suggested an IoT-based healthcare system model which mainly focused on patient monitoring and medical records management. Blockchain-based efforts on system scalability, traceability, and transparency in both centralized and decentralized models is suggested in [15–18]. In [15] authors proposed a privacy-preserving scheme based on blockchain and cryptography encryption techniques. Their work addresses the security and privacy issues; however, latency is not addressed properly during critical situations. The work in [16] proposed a medical system to verify the patient medical data integrity and authenticity at the edge system level, which eliminates the over-dependency on the third-party validation system. A blockchain-based patient medical record management system named MedChain is proposed in [17]. Their system successfully conducted the performance analysis with low latency and at high-level security. Authors in [18] suggested a smart healthcare framework that provides edge computing functionality for secure and smart health surveillance. Their framework exhibited scalability, privacy, and security issues. In [19] the authors presented a framework for a cloud-edge-based industrial internet of things that uses blockchain to secure and

protect data. An optimised consensus model for blockchain is suggested in [20]. Authors demonstrated how to reduce the overheads on edge-cloud paradigm.

In order to solve the issues of storing large data on a blockchain, many researchers provided various solutions. The work in [21] presented a modified version of IPFS with ethereum smart contracts to compensate for the blockchain data storage limitations. Authors in [22] introduced an edge based blockchain model to share the medical data over secure channels. Their research primarily concentrated on secure data transmission via Edge and blockchain. However, in this case the operational cost will be higher for continuous sensor data. In [23] researchers suggested a secure medical data sharing framework using edge and blockchain platforms where the motto is to reduce the latency and cost. The work is only limited to data sharing and access control. A next level framework is suggested in [24] where it offloads the data sharing and communication in a decentralised computing paradigm. Their work mainly designed to address the security and privacy issues of data offloading.

Researchers recently focused on integrating ML with IoMT/H-CPS to develop healthcare systems with intelligent decision-making capabilities. The authors in [25] used an IoMT framework for the heart disease prediction using modified salp swarm optimization (MSSO) and adaptive neuro-fuzzy inference system (ANFIS). To predict whether or not hypertension patients may experience hypertensive heart disease in the near future, a model known as XGB-SVM is used. Their method shows better performance compared to RF, Gradient Boosting Decision Tree (GBDT) and GBDT-SVM, where GBDT is used for feature selection and SVM used for classification on Area Under the Curve (AUC) and Receiver Operating Curve (ROC) metrics [26]. In their work XGBoost and SVM were used as feature extractors and classifiers, respectively. A method named χ^2-DNN was used for heart disease in [27], where χ^2 is used to eliminate irrelevant features. The work in [28] used a random search algorithm (RSA) for feature selection and a random forest (RF) algorithm for the prediction of heart disease. They have performed two types of experiments to evaluate their model: One with only RF model and another combined RSA with RF. Experiments are performed on Cleveland data set. The proposed method (RSA-RF) is efficient and less complex than RF as it generates 3.3% more accuracy. Here, they have selected 7 features from the Cleveland data set using RSA. In [29], researchers have suggested two regularized stacked SVMs for the diagnosis of heart failure. Here, the first $L1$ regularized linear SVM is used to discard irrelevant features. The second $L2$ regularized SVM with RBF kernel is used for prediction. All these works have demonstrated that ML based algorithms heart disease detection accuracy largely depends on selecting proper medical features.

Along with above works researchers focused on Blockchain integration with ML for classical healthcare systems [30–32]. A Blockchain-based fog computing framework and SVM classifier are used to develop a remote health monitoring system in [30]. However, their model increases the system latency when the feature size increases. The authors in [31] have defined a framework called Blockchain-Based Deep Learning as a-Service (BinDaaS) to keep privacy, confidentiality, and consistency of electronic health records (EHR) of patients in the remote server. They have used Deep Learning as-a-Service (DaaS) for disease prediction. However, they have not considered the latency of the model, which makes it hard for real-time implementation. Authors in [33] proposed privacy preserving model for cloud based clinical device support system. A privacy- assured fog-based data aggregation model is proposed in [34]. The works in [33, 34] are more focused on data privacy while it flow on Fog and cloud computing paradigms. Even today most of the systems are controlled and monitored by third-party or cloud service providers. In order to overcome these issues, we proposed a decentralised user centric blockchain-based healthcare system framework named Fortified-Chain [35] on a hybrid computing platform. Though it delivers EHR privacy and security, transparency, and availability by adopting blockchain, hybrid computing, and IoMT/H-CPS. however, IoMT/H-CPS is not intelligent enough for service automation. We observed that still there is a scope for improvement in the system scalability, availability and performance with the help of simplified decentralised access control mechanism on Hyperledger fabric platform with off-chain cache. The proposed Fortified-Chain 2.0 extends the system capabilities of [35] using additional features such as mutual identification, RFSVM based health status prediction and off-chain security with minimal latency in system operations.

IV. Architecture of Fortified-Chain 2.0

In this section, we briefly introduced the core elements used in the proposed system IV-A followed by its architecture, layers and workflow IV-B, IV-C and IV-D, respectively.

A. Preliminaries

1) Hyperledger Fabric and Chaincode

Hyperledger Fabric is a complete private blockchain platform that requires permissions to participate in blockchain operations. All of its smart contracts execute in separate Docker containers that prevent code overflow and interference with operations. The Fabric framework is built on four components such as certificate authority (CA), client, peer, and orderer. The CA is a certificate management authority, whereas the client is used for peer node interaction with user-end programs. The peer nodes in the fabric are categorized into two subcategories like endorsement (endorser) and transaction commitment (committer). The orderer node collects the transaction data from the peer nodes and adds the data to the distributed ledger.

2) Distributed Decentralised Storage System (DDSS)

The proposed blockchain-based peer-to-peer file sharing management system that fabricated IPFS with Hyperledger to form DDSS. Moreover, Distributed Hash Table (DHT) is used to store and retrieve the data from DDSS. In order to request file chunks from the appropriate hosting nodes using key-node values, a node first searches the DHT table for those files. When a device needs to access a specific file, it must first check its internal cache before reaching the DDSS, which

Table I: Features comparison with other frameworks

Work Name	Features	Computing Platform	Proposed Solutions	Automation	Off-chain Secure Cache
[12, 13, 15, 16],	Integrity, Security, Scalability	Cloud-based (Centralised)	Access Control mechanism	NA	NA
[14]	Network Virtualization	Hybrid computing (Decentralised)	security and Network virtualization	NA	NA
[17]	Access control, Privacy, Scalability	Cloud-based (Decentralised)	Smart-contract based Management System	Time-based smart-contracts	NA
[18]	Privacy, AI/ML based Decision Making	Edge-Cloud (Centralised)	Homomorphic Encryption	NA	Edge-level
[25–29]	Cloud and AI/ML	Cloud-based (Centralised)	Prediction and Classification techniques	Healthcare Prediction Applications	NA
[24]	Blockchain-based privacy ans security	Edge computing	Secure Data sharing mechanism	NA	NA
[36, 37]	Network virtualization	Hybrid computing (centralised)	EHR, Network management	General Healthcare Applications	NA
Fortified-Chain [35]	Privacy, Security, Traceability, Scalability	Hybrid (Decentralised)	Ring-based Selective Sharing	Event-based Smart-contracts	NA
Fortified-Chain 2.0 (Current Paper)	Privacy, Security, Traceability, Scalability, low latency	Hybrid (Decentralised)	Mutual authentication, Off-chain Security, Selective Sharing	AI/ML-based Smart contracts	Yes

reduces network load. The underlying IPFS system eliminates data redundancy by applying content addressing and version control mechanism.

3) Hybrid Computing

In order to improve the real-time data usage and to reduce communication latency, we integrated cloud and edge computing to establish a hybrid computing paradigm. Edge computing moves processing and storage capabilities closer to the originating source resulting in quicker reactions and real-time services. For decentralised applications, hybrid computing offers a number of advantages over the cloud model including low latency, no SPoF, and no third party control over data. The edge computing has drawbacks like restricted process and storage capabilities. On the other hand, cloud computing offers enormous processing and storage capacity but lacks in real-time data analytics. Hybrid computing combines the benefits of both technologies to achieve low latency, scalability, and decentralised data management.

4) Modelling Techniques and Machine Learning

In machine learning, both RF and SVM are supervised learning algorithms, where RF is an ensemble bagging type of machine learning algorithm [38]. The working principle is that it takes n subsets of samples from the training data with replacement and trains n number of decision trees with each subset of data. Moreover, it selects n subsets of feature vectors from the feature set without replacement and uses those subsets of features to train each decision trees (DTs). Finally, to decide the output class of a sample, it performs voting on the outputs of the DTs and selects the class with the majority of votes as given in 1.

$$v = argmax\left(\forall c \sum_{i}^{m} (\hat{o_i} == c)\right) \tag{1}$$

Where, $\hat{o_i}$ is the output of the i th DT and c is a class type. Given, a set of training examples, the SVM classifier maps the training samples into a hyperspace so that it maximizes the separation boundary between two or more categories. During the test, new samples are mapped into that same space and predicted to a category based on which side of the gap they fall. The problem formulation of the SVM classifier is as follows in 2.

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{p=1}^{P}\xi_i \\ subject \quad to \quad & y_i(w^T\phi(x_i) + b) \geq 1 - \xi^i, \\ & \xi_i \geq 0 \end{aligned} \tag{2}$$

Here, w, b, ξ are the weights, bias, and slack variables. The performance of the SVM classifier depends on the SVM kernel, Gamma (γ), and the C parameter. The kernel function defines the classifier boundary. There exist three types of kernel functions i) linear, ii) polynomial, and iii) Radial Basis Function (RBF). In the case of the linear kernel, the decision boundary is linear. In contrast, in a polynomial kernel, the decision boundary is a curvature, and the curvature depends on the degree of the polynomial. For the RBF kernel function, the decision boundary is formed by a set of Gaussian functions. The Gamma and the C parameter in the SVM model is used to control the variance-bias dilemma in the system.

B. Architecture

In this section, we discussed the Fortified-Chain 2.0 architecture and its modules followed by the system-specific mechanisms. In addition, we suggested Fortified-Chain 2.0 with different views in the sub-section IV-B, and its flow work in the sub-section IV-D. Finally, we presented the cryptography mechanisms of the system in the sub-section V-A. The Fortified-Chain 2.0 consists of three logical layers named as Data Layer, AI/ML Layer, and Alert/Automation Layer illustrated in Fig.1. Further, the layered view is divided vertically into two logical layers, such as hospital internal and hospital external service networks. The layered design enables administrators to apply privacy and security methods on a specific layer without interfering with the functionality of

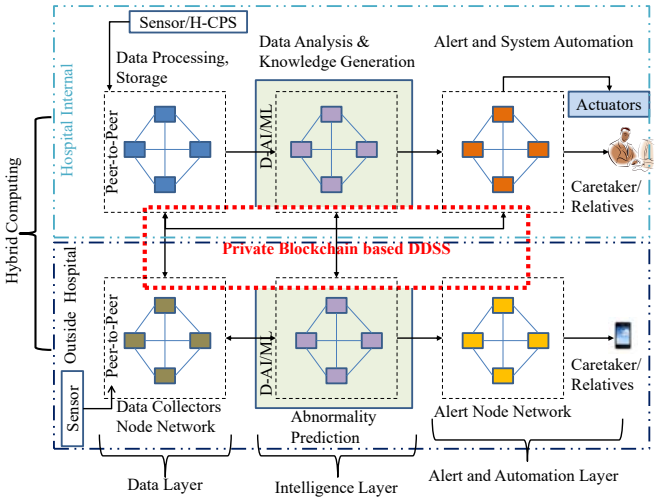


Figure 1: The Layer-view of Proposed System

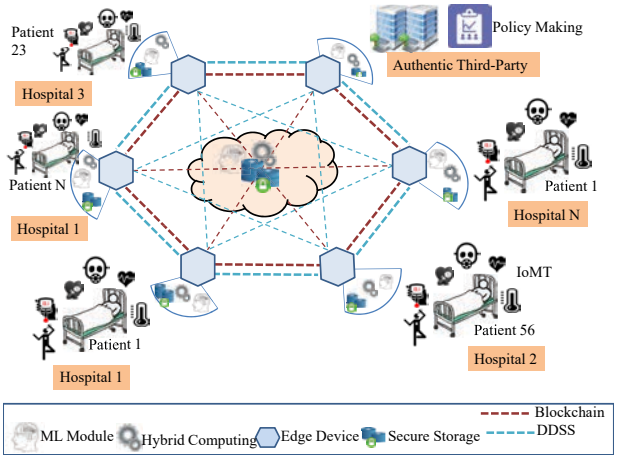


Figure 2: Overview of the Proposed System Architecture

other layers. The proposed access control mechanism prevents unauthorised remote layers to safeguard the local resources. For instance, the data layer elements are unable to access the resources of the AI/ML layer or alert/automation layer without proper authorization. Moreover, each layer operates in a different virtual environment, preventing it from influencing the operations of other layers. Every layer has a unique gateway for communication. These gateways are selected in such a way that there are always enough gateways available to spread the communication load. The proposed system conducts an election among all the nodes that are connected to hybrid computing. Based on the node influence, hybrid computing distributes the roles and responsibilities to the nodes. During the node selection process, hybrid computing considers the node's awake time and its identity before allocating it to a critical job. The selection of gateways is based on the edge node and its follower count. The edge node with more IoMT followers is given a higher priority to become a gateway.

C. System Architecture Layers

1) Data Layer

The data layer consists of sensors (data generators), edge computing nodes (used to process data), and storage network (DDSS), respectively. In this layer, hybrid computing performs all data management tasks for the proposed system. A dedicated group of nodes performs decentralized data storage and management using IPFS and blockchain. in order to achieve the DDSS operations faster, every gateway maintains a secure off-chain ledger. The cryptography operations of selective sharing (SSM) are also applied on this layer to provide better privacy and security of personal medical data.

2) AI/ML Layer

It is logically separated into two sub-layers known as hospital internal layer for in-patient services and hospital external layer for remote patient services. The internal intelligence layer performs real-time data analysis and heart disease prediction for internal patients. On the other hand, the external layer carry out the same tasks to serves remote patients with the help of cloud computing. The proposed RFSVM algorithm helps to identify patient cardiac issues in advance.

3) Alert/Automation Layer

The alert/automation layer of Fortified-Chain 2.0 is the bottom layer, where alerts are produced using the information from the intelligence layer. This layer consists of actuators, system terminals, and healthcare robots. It has separate modules for internal and external hybrid computing as shown in Fig. 1.

4) Hospital Internal Hybrid Computing

For simplicity and easy management, the total system operations are logically split into two parts. The internal operations of the hospital are handled by edge computing (also an integral part of the hybrid computing paradigm), which also offers intelligence, processing, and storage capabilities.

5) Hospital External Hybrid Computing

External hybrid computing completely depends on the cloud computing for system operations. It is logically connected to the hospital's internal hybrid computing by employing blockchain and DDSS. The Fig.2 represents the overview of the Fortified-Chain 2.0 where every stakeholder uses it for medical service purpose.

D. Workflow

The internal modules of Fortified-Chain 2.0 are depicted in Fig. 3. In our proposed module sensors generate and transfer patient health information to the nearest edge computing node for processing and analysis purpose. We considered the following use-case for better understanding. At first, patients are enrolled with the help of a hospital registration and record management (HRRM) system, which includes a specialized healthcare cyber physical system (H-CPS). The processed data is then uploaded to the DDSS and local cache as per the predefined access rules. Now, the internal hybrid computing uses the local cache and predicts the disease. Based on the

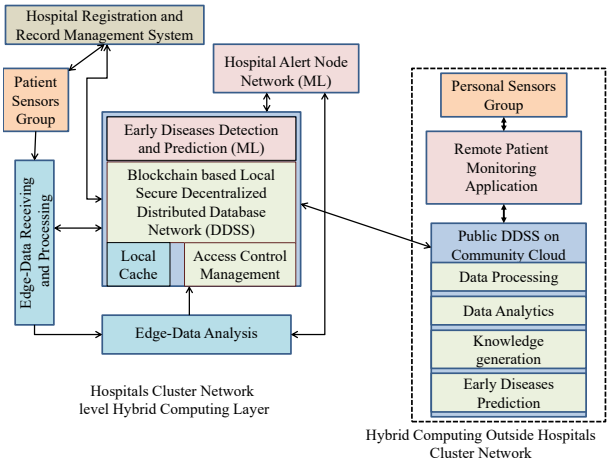


Figure 3: Overview of Proposed System's Internal Modules

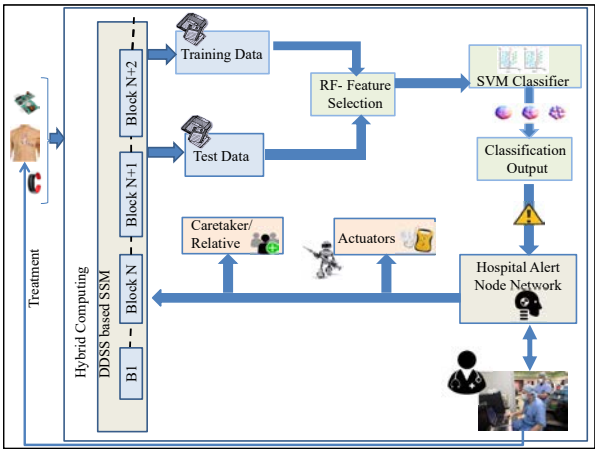


Figure 4: Architecture of the ML Module

predictions, alerts are sent to the respective doctors and healthcare providers with the help of the alert/automation layer. The predictions and alerts are dynamically generated based on the current local cache data. Every action and event is written to the DDSS for transparency and traceability purposes. The system utilizes the cloud-based hybrid computing to provide remote services and there after the patient information is directly sent to the DDSS. The processed data is passed to the ML prediction module to predict and generate alerts for the patients as shown in Fig. 4. The proposed RFSVM ML model consists of RF and SVM to predict the heart disease of a patient in real-time DDSS.

V. Proposed Algorithms for Fortified-Chain 2.0

A. Algorithms and Mechanisms

The proposed security mechanism consists of mutual authentication and a selective ring-based access control system. Over a finite field of size 263, the hybrid computing system selects an elliptical curve $y^2 = x^3+2*x+3$ [39]. At first, hybrid computing generates a secret number (X_{HC}) using the true random number generator method, which produces a private key. The hybrid computing public key is generated using the point doubling method with a fixed generator point as show in Fig.5. The left hand side of the figure represents the two points

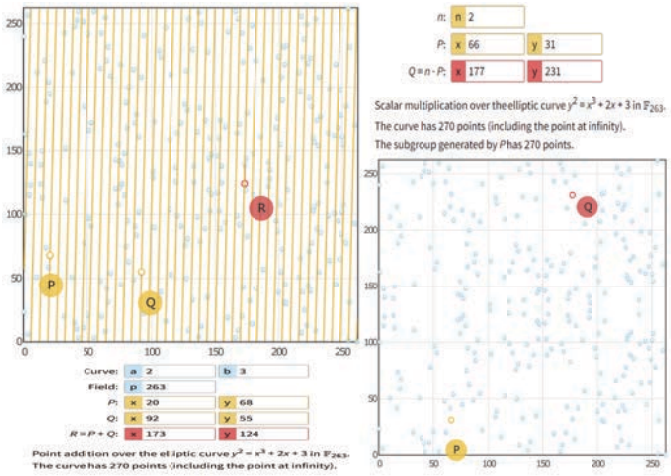


Figure 5: The Overview of ECC operations

addition to generate new secret point Q. On the other side, it represents the multiplication point to generate secret point Q. The objective of mutual authentication algorithm is to register and validate the node's authenticity. Also, mutual authentication eliminates the hybrid computing interference during node-to-node secure communication. The two nodes negotiate and generate one session key for future secure communication. Mutual authentication enables the authorised nodes to bypass the hybrid computing-based secure session key exchange that eliminates computational load on edge computers. The node identity and mutual authentication algorithm further logically divided into three subsections where the first two subsections represent the node registration and identity generation (SID_{node}, MAT_N) and the third subsection constitutes for mutual authentication and authorization among nodes.

The overall flow of the proposed node identity and mutual authentication algorithm 1 starts with the node (sensors and actuators) registration. Edge computer receives requests from sensors and actuators, and generates different node identities as show in step (1) in flow diagram 6. During the node registration process (2), each node shares their node type (N_{type}) and public key (Pub_N) with the edge computer for further communication. Simultaneously, the edge computer acknowledges the communication and generate registration identity (R_{ID}) for each node in step (3). The registration identity (R_{ID}) is protected with node's public key (Pub_N) encryption. In step (4), edge computer computes and stores H(R_{ID}) and R_{ID} in its cache, and concurrently generates the group identity (G_{ID}) as well as accumulates in local database. The registration process concludes with the dispatch of R_{ID} to the requester node. The node initiates the next level of communication with the edge computer using R_{ID} and G_{ID} to get the service identity (SID). In the next step (6), the edge computer generates a mutual authentication token (MAT_N) for each registered node. The edge computer transfers the SID and MAT_N to the respective node in an encrypted format with the help of the node's public key. In future, these nodes utilize their SID, MAT, and their public key to establish a secure communication with other nodes. The final subsection validates the node-to-node mutual authentication in an abstract form. Here, the nodes establish

Table II: Notations and detailed description

$Add()$: new rule generator	$actor_{ID}$: actor identity	E_{ID} : device embedded identity
FN_{req} : requested file name	FL_{acc} : file access code	FL_{req} : requested file
SID_N : service identity	Nisynch: claim of man in the middle	FL_{add} : file address on DDSS
G_{ID} : group identity	ID_{hos} : hospital identity	Niagree : claim of denial of service
$Gen()$: identity generation function	ID_{div} : device identity	ID_{rin} : ring identity
P_{ID} : patient identity	DOC_{ID} : doctor identity	Pub_{actor} : requester public identity
Pub_N : node public identity	Pub_{edge} : edge identity	PU_{div} device public key
$P()$: permutation function	Prv_{edge} : edge private key	R_{ID} : registration identity
P_{ID} : patient identity	PR_{edg} : private key of edge computer	MAT_{node} : mutual authentication token

Algorithm 1 Node identity and mutual authentication

Input: $Req[N_{type}, Pub_N]$
Output: Validate Node Authenticity

```
/* Node registration */
```

if $Req[N_{type}, Pub_N]$ *is Valid* **then**
- $Gen(N_{type}, Pub_N)$
- Edge: Ack $\rightarrow$ Node
- Edge: generate R_{ID}, G_{ID}
- Edge: $H(R_{ID})$, G_{ID} $\rightarrow$ local database
- Edge: $E(Pub_N, [R_{ID}, Pub_{edge}, G_{ID}])$ $\rightarrow$ Node
- Node: $D(Prv_N, [R_{ID}, Pub_{edge}, G_{ID}])$ $\rightarrow$ cache

else
- Do the communication reset

```
/* Getting MAT */
```

if *Both nodes are valid* **then**
- $E(Pub_{edge}, E(Pub_{edge}, P_{ID})||H(R_{ID}))$ $\rightarrow$ Edge
- Edge: $D(Prv_{edge}, D(Prv_{edge}, P_{ID})||H(R_{ID}))$
- **if** $(P_{ID})||H(R_{ID})$ *is Valid* **then**
  - $SID_N = P(P_{ID} \oplus R_{ID} \oplus G_{ID})$,
  - $MAT_N = SID_N \oplus P(E_{ID})$
  - $E(Pub_A, (SID_A, MAT_A))$ $\rightarrow$ Node A
  - $E(Pub_B, (SID_B, MAT_B))$ $\rightarrow$ Node B
  - Node A $\leftarrow Store(P_{ID}, R_{ID}, SID_A, MAT_A, Pub_{edge})$
  - Node B $\leftarrow Store(P_{ID}, R_{ID}, SID_B, MAT_B, Pub_{edge})$
- **else**
  - Do the communication reset.

else
- Do the communication reset.

```
/* Nodes mutual authentication without Edge computer */
```

if *Both nodes allowed to comunicate independently* **then**
- $E(Pub_B, [SID_A \oplus MAT_A||Pub_A])$ $\rightarrow$ Node B
- `/* At Node B */`
- Node B: $D(Prv_B, [SID_A \oplus MAT_A])$
- $P(E_{ID})_A = SID_A \oplus MAT_A$
- **if** $H(P(E_{ID})_A) == H(E_{ID})_B$ **then**
  - Node B: $Token$|| Request for $Data$ $\rightarrow$ Node A
  - Node A: $AccessCode$+$EncrptedData$ $\rightarrow$ Node B
- **else**
  - Do the communication reset.

a secure communication without the interference of an edge computer. The recipient $Node_B$ validates the identity of sender node ($node_A$) by comparing the sender's $P(E_{ID})$ identity with its gateway edge identity. All the nodes under a single hospital have the same and unique $P(E_{ID})$ to differentiate

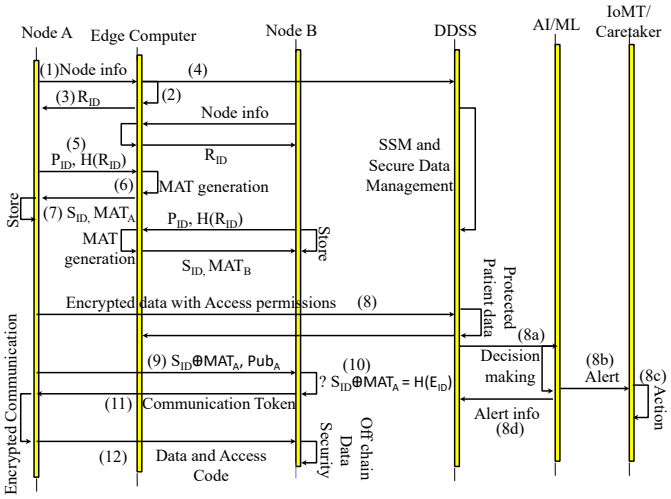


Figure 6: The Overview of Operational Flow

from other nodes in the Fortified-Chain 2.0.

The following subsections represent the overview of operations considering from ring generation to SSM based request validation in a step wise format.

1) Data upload and validation process

i. The sensor encrypts the medical data of patients by using the following operation as shown in 3 and transmits it to the edge computer.

$$\begin{aligned} x &= E(SID_A, PatientData); \\ x' &= x \parallel SID_A; \\ y' &= (Timestamp \parallel Token); \\ E(Pub_{edge}, &\left(x' \parallel Hash\left(PatientData\right) \parallel y'\right); \end{aligned} \tag{3}$$

ii. The sensor encrypts the sensed data and appends with its group and service identity. Concurrently, it generates a temporary token to acknowledge the communication with other nodes. For every successful communication the token value is incremented by 1. The data from sender node is protected by encryption with the help of edge public key. The fully transferred data structure is as follows: $E(Pub_{edge}, (R_{ID}||SID||Data||MAT_A))$, where $Data$ can be represented as $[P_{ID} \oplus SID \oplus Token]$.

iii. The edge computer receives request from the sensors and performs SID and P_{ID} validation to check the DDSS access permissions. The edge computer decrypts the received data and validates its integrity before writing it to DDSS. The received data comes along with immutable user-defined access rules that enables no third party control over the data. Additionally, the edge computer

maintains the shared data confidentiality on off-chain network storage with the help of user specific encryption key, which is used as a base key for generating access codes. The off-chain data helps the system to speed up the data retrieval and utilization.

iv. The edge computer responds to file upload requests from different nodes and generates a new $Ring_{ID}$ for every sensor data. This event occurs when the data hash does not exist in the access rule table of the SSM. An acknowledgement $E(SID, (Token \oplus P_{ID}||Ring_{ID}))$ is sent to the node as response.

v. Finally, AI/ML modules of the edge computer fetched data from DDSS is decrypted and validated its integrity before considering it for decision making.

2) Ring generation process

The ring generation algorithm 2 outlines the overall process of new ring generation in lookup table.

Algorithm 2 Ring Generation

Input: $Data_{Id}$, $AccessRules$, $actor$, $level$ and P_{ID}
Output: New Ring
Function $RingGenerate()$**:**
  Edge: Read input parameters
  Edge: $SetPermissions(Data_{Id}, AccessRules, actor, P_{ID})$
  Edge: Add($Data_{Id}$, G_{ID}, $actor_{ID}$, $Permissions$, P_{ID}, $level$) $\rightarrow$ lookup table
  Edge: Update DDSS ledger
  Edge: Update cache
  return New Ring details $\rightarrow$ Patient

i. The function $RingGenerate()$ generates new rings for currently joined patient data using parameters such as $AccessRules$, $actor$ and P_{ID} respectively. The given inputs are useful when SSM needs to validate the request against access permissions. The edge computer initiates a new ring generation process by sending a notification to the user when no rules are defined in the lookup table. Upon user's approval a new ring is generated with approved access permissions by the function $SetPermissions()$ call and adds the new ring to DDSS by employing $Add(G_{ID}, actor_{ID}$, Permissions, $P_{ID}, level)$ function call.

ii. When a request is originated from the same hospital network, then the edge local cache is used to validate requester access permissions. However, if the request is from a remote hospital, then edge takes the help of DDSS and SSM for fetching the global level access control rules to validate requester access permissions.

iii. Every time the edge computer validates requests against lookup table rules and updates newly generated rings of the lookup table. This process helps the requester and edge to minimize the latency caused by DDSS-based searching. The lookup table is only used for the first time validation; later on edge local cache is used for faster response.

iv. The hybrid computer protects lookup table integrity and confidentiality on DDSS by allowing only authorized nodes to access it. Blockchain smart contracts are used to filter the request based on the lookup table rules and hybrid computing business logic.

3) Access control and Selective sharing Mechanism

i. Based on the patient's instructions the edge computer generates SSM access control rings on DDSS for their data. The ring consists of access rules for actors and devices, which list out all possible data operations for that group. Further, the ring contains dynamic data deryption access codes which are hidden from public visibility.

ii. The ring provides dynamic access codes to different actors based on their access permissions for the requested data. The dynamic access codes are derived from patient's secret key using ECC curve. The rules are stored on DDSS with a ledger name lookup-table, also a copy of that ledger is maintained in edge's local cache.

iii. A true random number generator function creates the ring identity ($Ring_{ID}$) for every new ring. Every ring and it's $Ring_{ID}$ is unique so it is easy for finding a ring in ordered lookup table on DDSS.

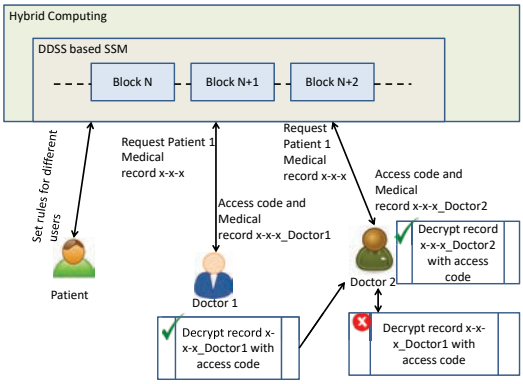


Figure 7: The Overview of Data Protection at User-end

The SSM provides off-chain security to patient medical records by employing requester-specific access control codes and dynamic medical data copies. It locks the copies with receiver device identity and requester identity. The non-authorized recipient first needs to get an access code and specific device along with the requester identity to decrypt the data. The possible actions on off-chain data are presented in Fig. 7. Doctor 1 has access to Patient 1's medical record at DDSS. Doctor 1 received the data after successful request validation. When doctor 2 wants to read the same data by making a copy from Doctor 1's device, the offline security on Docter 2's device reads the identity of the device before starting the decryption process of the files. Because the device is different, off-chain security prevents the decryption process. In another case, if doctor 2 tries to decrypt it on the same device, doctor 2 should have access to the decryption code.

B. Proposed ML Algorithm

In this work, we proposed a Fortified-Chain 2.0 decentralised ML module called RFSVM. This module consists of two different ML algorithms, RF and SVM, where RF acts as a feature selector and SVM works as a classifier. We initially trained the RF algorithm using a heart disease data set named

Cleveland and then computed the importance of each feature present in the heart data set by adopting the Gini index. Later, we sorted the features based on the Gini Index and selected a subset of important features from the original feature set based on a threshold value ($\Im$). Finally, the selected subset of features are applied to the SVM classifier to identify whether a patient is prone to have a heart disease or not. The whole process has been illustrated in algorithm 3 and algorithm 4. Algorithm 3 talks about how the data has been received by the Fortified-Chain 2.0 system. Whereas, the algorithm 4 explains the complete working principle of the decentralised RFSVM module.

In this work, we deployed this Fortified-Chain 2.0 decentralized RFSVM modules to automate primary health tasks by taking real-time DDSS data stream. The proposed model helps the healthcare service providers in servicing more patients in limited time and resources with the help of accurate data and primary diseases classification from ML modules.

Algorithm 3 Read input dataset

Input: Input dataset in csv format.
Output: Feature vector and class labels.
Function `readFile()`**:**
- Read input dataset using pandas.read_csv() into a data frame
- Shuffle the dataset
- Split input dataset into features and class labels.
- Scale the feature vectors using z-score normalization
- **return** Feature vectors, Labels

Algorithm 4 RFSVM Classifier

Input: Feature vectors, Class labels
Output: Test accuracy
Function `classifySVM()`**:**
- Define K, Gamma, C and Type of SVM_kernel.
- Define *test_accuracy* = []; i = 0;
- **while** `i<K` **do**
  - Find split positions for current fold.
  - Split feature vectors into training and validation set.
  - Split class labels into training and validation set.
  - Initialise a Random-Forest (*RF*) Classifier.
  - Train *RF* on training set.
  - Extract important features based on threshold of *RF*.
  - Create a pipeline of steps containing SVM Kernel Type
  - Define parameters = [C, Gamma]
  - Instantiate SVM Model for [pipeline, parameters]
  - Train model on important-features set.
  - Test model on validation set.
  - Store result in *test_accuracy*.
  - i = i + 1;
- Compute average accuracy of all folds.
- **return** avg_test_accuracy

VI. Experimental Results

We illustrated test setup, machine learning accuracy and system performance of our proposed system in the subsection VI-A, VI-B and VI-C, respectively. We also compared our proposed system performance with two existing works such as fortified-chain and H-CPS system. Finally, the subsection VI-D discusses the system security analysis.

A. Environment

We prepared our test setup as shown in Fig.8 to find its real-time behaviour. We evaluated the proposed Fortified-Chain 2.0 by considering a locally configured Hyperledger Fabric 1.4

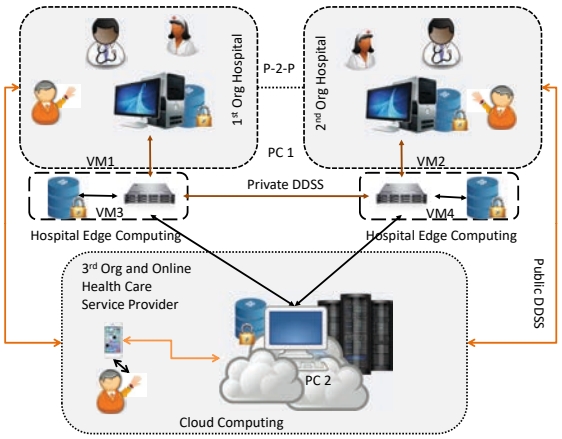
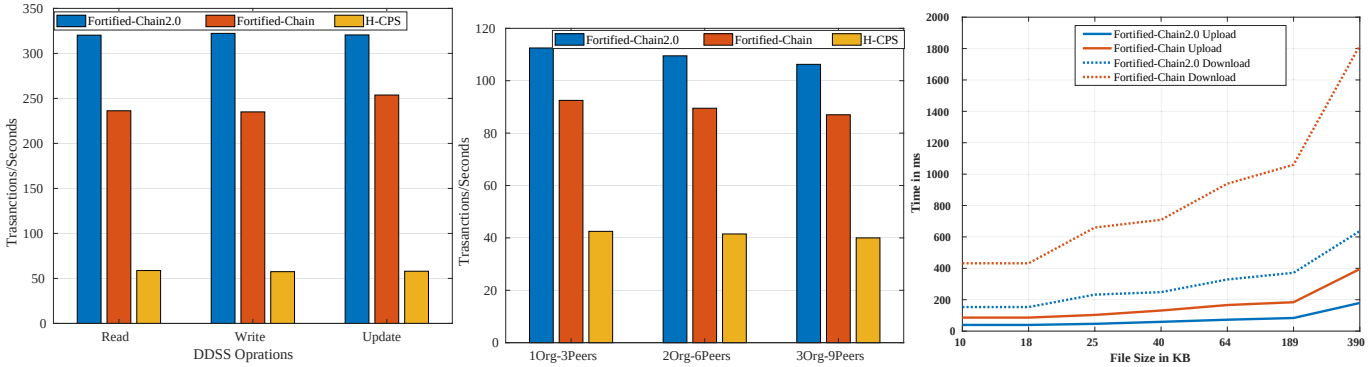


Figure 8: Experimental setup for file operations simulation

```
CONTAINER ID  IMAGE  COMMAND  CREATED  STATUS  PORTS  NAMES
94161f65ef91  dev-peer0.centauth.ehr.com-ehrcc-1.0-
d0e17ea4ad50c8d7dac629dc6e0359469ed946b9ac5fab72389b4dceb4c25a2b  "chaincode -peer.add…"  9 days ago  Up 9
days                                                                dev-peer0.centAuth.ehr.com-ehrcc-1.0
f83619174810  hyperledger/fabric-tools "/bin/bash"  9 days ago  Up 9 days  cli
567f06357f10  hyperledger/fabric-peer  "peer node start"  9 days ago
Up 9 days  0.0.0.0:21051->7051/tcp, :::21051->7051/tcp, 0.0.0.0:21053->7053/tcp, :::21053->7053/tcp
peer0.healthCareProvider.ehr.com
884012d7b7b6  hyperledger/fabric-peer  "peer node start"  9 days ago
Up 9 days  0.0.0.0:9051->7051/tcp, :::9051->7051/tcp, 0.0.0.0:9053->7053/tcp, :::9053->7053/tcp
peer0.radioLogist.ehr.com
bc038b9eaf51  hyperledger/fabric-peer  "peer node start"  9 days ago
Up 9 days  0.0.0.0:20051->7051/tcp, :::20051->7051/tcp, 0.0.0.0:20053->7053/tcp, :::20053->7053/tcp
peer0.researcher.ehr.com
541410f3ea5d  hyperledger/fabric-peer  "peer node start"  9 days ago
Up 9 days  0.0.0.0:10051->7051/tcp, :::10051->7051/tcp, 0.0.0.0:10053->7053/tcp, :::10053->7053/tcp
peer0.pharmacist.ehr.com
ad615f914a7c  hyperledger/fabric-peer  "peer node start"  9 days ago
Up 9 days  0.0.0.0:8051->7051/tcp, :::8051->7051/tcp, 0.0.0.0:8053->7053/tcp, :::8053->7053/tcp  peer0.clinician.ehr.com
a2fc9826ecd4  hyperledger/fabric-peer  "peer node start"  9 days ago  Up
9 days  0.0.0.0:7051->7051/tcp, :::7051->7051/tcp, 0.0.0.0:7053->7053/tcp, :::7053->7053/tcp  peer0.centAuth.ehr.com
8990f3b7a2bd  hyperledger/fabric-couchdb  "tini -- /docker-ent…"  9 days
ago  Up 9 days  4369/tcp, 9100/tcp, 0.0.0.0:9984->5984/tcp, :::9984->5984/tcp  couchdb4
a2cc66a83935  hyperledger/fabric-couchdb  "tini -- /docker-ent…"  9 days
ago  Up 9 days  4369/tcp, 9100/tcp, 0.0.0.0:10984->5984/tcp, :::10984->5984/tcp  couchdb5
cada6d719a46  hyperledger/fabric-orderer  "orderer"  9 days ago  Up
9 days  0.0.0.0:7050->7050/tcp, :::7050->7050/tcp  orderer.ehr.com
dbc9697b8e8c  hyperledger/fabric-ca  "sh -c 'fabric-ca-se…"  9 days ago
Up 9 days  0.0.0.0:8054->7054/tcp, :::8054->7054/tcp  78868b0dc389  hyperledger/fabric-
couchdb  "tini -- /docker-ent…"  9 days ago  Up 9 days  4369/tcp, 9100/tcp,
0.0.0.0:6984->5984/tcp, :::6984->5984/tcp  couchdb1
```

Figure 9: Fortified-Chain 2.0 Hyperledger docker images at run-time

platform with five virtual systems. Two systems with 4GB RAM are configured and dedicated for edge computing, and one system with 8GB RAM and 50GB storage is assigned for cloud node. Two more systems with 4GB RAM each used as organizational level or user personal level device use-cases. We considered publicly available medical records of [40] various sizes to validate the DDSS operational behaviour. All inputs to the DApp application is parsed using XML parser. All testbed devices are connected with the help of Blockchain, IPFS, and Docker swarm. The main objective of the experiment is to demonstrate the performance of the proposed Fortified-Chain 2.0 scalability, transparency and data availability without compromising the data security and privacy. We considered CouchDB state database for users medical ledger management at each organization. Fig. 9 represents the respective docker images and CouchDB databases. All VMs are configured on Intel i5 CPU with 2.8 GHz and 64 bit Ubuntu 16.04 operating system. We considered React and NodeJs platforms to develop user-end API. The smart contracts are initialized and accessed using the fabric platform. We used the Cleveland data set [41] to evaluate the ML model's performance.



(a) Data Operational Throughput (b) System scalability respect to Nodes Count (c) System Response respect to File size

Figure 10: A Selected Experimental Results of Fortified-Chain 2.0

B. Classification Accuracy and Performance

To evaluate the performance of the proposed RFSVM model, we compared the proposed model with the Naive Bayes (NB), RF, SVM, and Method (RSA and RF) in [28]. The authors in [28] used RSA for feature selection and RF for classification. Further, to evaluate the feature selection capability of the RF module, we have compared the proposed RFSVM model with PCA and SVM, χ^2 and SVM, Lasso and SVM, and XGBOOST and SVM (XGB-SVM) [26], where PCA, χ^2, Lasso and XGBOOST worked as feature selector, respectively. Here, ten-fold cross-validation is used to obtain the classification results. We have used the same seed value for cross-validation in all the comparing algorithms. For SVM classifier, we have used a set of C values {0.01, 0.1, 1, 10, 100, 1000}, a set of γ values {0.0001, 0.001, 0.01, 0.1, 1}, and RBF kernel function. We illustrated the simulation results in Table III, where the boldfaced entry shows the best performance in the table III. From the experimental results, we can observe that the performance of the proposed RFSVM is better in comparison with the SVM classifier, which ensures the importance of the feature selection by the RF module. Similarly, the performance of the proposed model is better than the RF classifier, which shows the importance of this hybrid RFSVM model. Furthermore, the simulation results show that the performance of the proposed RFSVM model is better than other feature selection methods including two recent works: Method (XGBSVM) [26] and Method (RSA and RF) [28].

Table III: Accuracy of the Cleveland data set on different classifiers

Model	Accuracy
RF	85.46
NB	83.80
SVM	87.79
PCA - SVM	81.44
χ^2 - SVM	82.15
Lasso - SVM	87.13
XGB - SVM [26]	88.13
RSA - RF [28]	85.95
RFSVM	**89.12**

DDSS Functions	Computation Cost
DDSS Hash Function	6.3 ms
DDSS Transaction Decoding	20 ms
ECC Encryption	12.2 ms
ECC Decryption	5.3 ms

Table IV: DDSS functions cost test

C. Fortified-Chain 2.0 Performance

We performed simple file operations like records reading, writing, updating on the DDSS system. Simultaneously, we imposed access control and identity management tasks to check the system scalability for different scenarios and use-cases. We performed concurrent operations in VM1, VM2, and the user-end devices for the experimental evaluation. Fig. 10a illustrates the system's overall transaction throughput capability when compared to the Fortified-Chain [35] and raw blockchain-based H-CPS. The proposed system performed comparatively better than the existing Fortified-Chain and H-CPS due to less number of SSM operations required for data read, write and update on DDSS. The Blockchain-based H-CPS model is under performed when compared with Fortified system model due to resource constraints and heavy blockchain management operations. We considered different nodes and peer sizes to demonstrate our proposed system's scalability in real time by comparing with its predecessor as shown in Fig.10b. The present model is consistent in performance because of reduced operational load on edge nodes compared with earlier Fortified-Chain[35]. We also computed time required for file uploading and downloading through our proposed access control and mutual identification mechanism with Fortified-Chain mechanism. Fig. 10c illustrates the observed time in millisecond (ms) for each file size. The response rate is better in our proposed work by introducing the edge level cache. We presented DDSS functions cost analysis in Table IV, the DDSS functional cost are minimal so it is not overloading or adding huge delays in the system operations. The local cache is refreshed every 10 minutes for non-essential apps and every minute for crucial operations using the IPFS file version control system. By altering the DDSS update rules, even the time periods for non-critical tasks may be extended. We used the SHA-256 hashing algorithm with IPFS file version management

mechanisms to track changes in the local cache. Before being retrieved from the cache, the hash of the data is compared to the hash value kept on the DDSS. When the local cache loses its integrity as a consequence of any modification, the IPFS version control mechanism fetches a copy of the data and saves it as an updated cache. According to Fig.10, Fortified-Chain 2.0 outperformed the original Fortified-Chain in terms of throughput, scalability, and latency. Also, Fortified-Chain 2.0 reduces the burden on IoMT/H-CPS to comply with the system-level security.

D. Security Analysis and Discussion

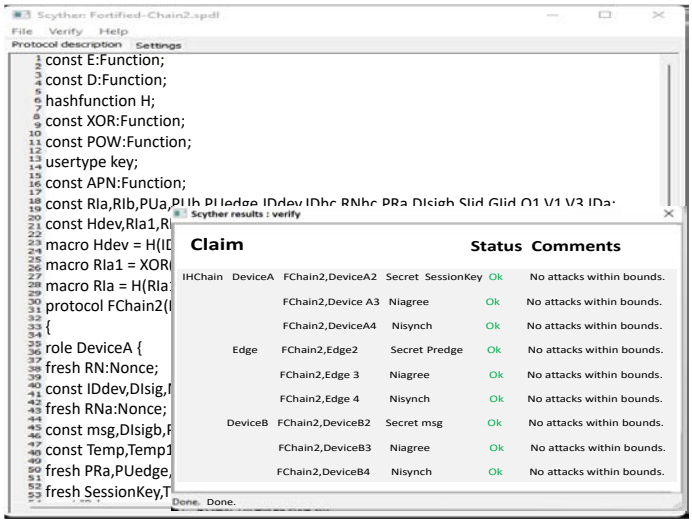


Figure 11: Scyther Tool Protocol Analysis Results

We presented system security analysis and protocol validation of the proposed architecture by considering suitable threat scenarios. We used the Scyther tool to validate the proposed system communication protocol. We considered two IoMT devices and one edge device use-case for validation. From the tool validation it is obvious that no information leakage is identified as shown in Fig. 11. In addition to Scyther, we considered one more tool named Microsoft thread modelling tool (MTMT) to discover relevant threats connected to the proposed architecture. The MTMT generates a list of prospective threats via the spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege (STRIDE) approach. The tool suggested the cache and data spoofing, data tampering, actuator configuration manipulation, non-repudiation, information exposure at the storage and transmission levels, and lastly escalation actress privileges are possible risks and needs mitigation mechanisms. Additionally, as shown in Fig. 12, we used the AT-AT Attack Tree Analysis Tool to analyse the proposed system for internal attacks. According to the AND gate, a parent node is only true when all of its children are true. The OR gate indicates that the parent node is also true if any child node is true. We have considered all possible attacks at leaf level nodes. To access the encrypted and hidden cache from the edge and user devices in path 12 $\rightarrow$ 6 $\rightarrow$ 5$\rightarrow$ 4$\rightarrow$ 3$\rightarrow$ 2$\rightarrow$ 1, the attacker must get access codes from DDSS, which will be only provided with authorised parties. The brute force attacks are stopped in the beginning stages due to SSM based access control and authentication methods. Because the attacker does not know the business logic for selecting the private points on the ECC curve, brute force attacks fail to crack the keys. On path 20 $\rightarrow$ 19 or18 $\rightarrow$ 15$\rightarrow$ 14$\rightarrow$ 1, if an attacker wants to compromise privacy by gaining access to some protected cache, then they must first gain access to the edge cache and access code which are encrypted and can only be decrypted by authorised parties. So even bypassing the SSM with a fake identity is a very expensive and time-consuming task because the attacker must find the secure points on the ECC curve. The access controls in the SSM mechanism completely regulate the other two avenues that might compromise privacy. Pseudo-identification numbers are used to safeguard the original personal identities, and the keys are created by multiplying random integers on the ECC curve. Since the mapping logic from personal identification to pseudo identity is a one-way function, a hacker cannot construct the actual identity using a pseudo identity. The only viable attack path for the attacker on the system is 35 $\rightarrow$ 30 or 31 $\rightarrow$ 27 $\rightarrow$ 26 $\rightarrow$ 1, in which instance the attacker needs to compromise two thirds of the devices in order to make the system unstable. However, the PBFT with IPFS node management system aids in the DDSS's ability to keep adequate copies of the information available for the user. Because no one may alter smart contracts after they are broadcast to the network, the path from node 35 to node 26 cannot be compromised. The last route from node 36 to node 26 is completely invalid since it requires the attacker to change the IPFS file system's source code rules, which is impossible in active networks. According to the study mentioned above, our DDSS can survive potential assaults, making the system reliable and safe. The Fortified-Chain 2.0 guarantees data integrity and establishes transparency in data utilization applying DDSS and SSM mechanisms. Because of event public ledgers on DDSS, no actor or device can not deny its activities. Using the sender's and receiver's ECC public and private key pairs, an encryption and decryption process is used to preserve the data confidentiality. A point-doubling technique generates access codes for sharing. The SHA-256 hash method is used to track the DDSS data integrity together with the IPFS file version control mechanism. The SSM uses hash table to map the data with its users that prevents the backtracking attacks on DDSS. SSM hides the original identities of its users and devices in access control table. Therefore, it is impossible for intruder to assume the real identity of victim from hash string. From the compilations the proposed Fortified-Chain 2.0 is achieved satisfactory level security in order to handle real-time threats. The corresponding reports are available at [42].

VII. Conclusion

Our proposed model addressed security and privacy issues in decentralized smart healthcare systems with the help of SSM. The hybrid computing paradigm solved well-known problems such as high latency and SPoF in classical cloud-centric models. The selective ring-based access control mechanism with mutual authentication responsible for data ownership, privacy, and security management. In addition, a tamper-proof public ledger guaranteed for transparency in DDSS and also responsible for system operations. The proposed

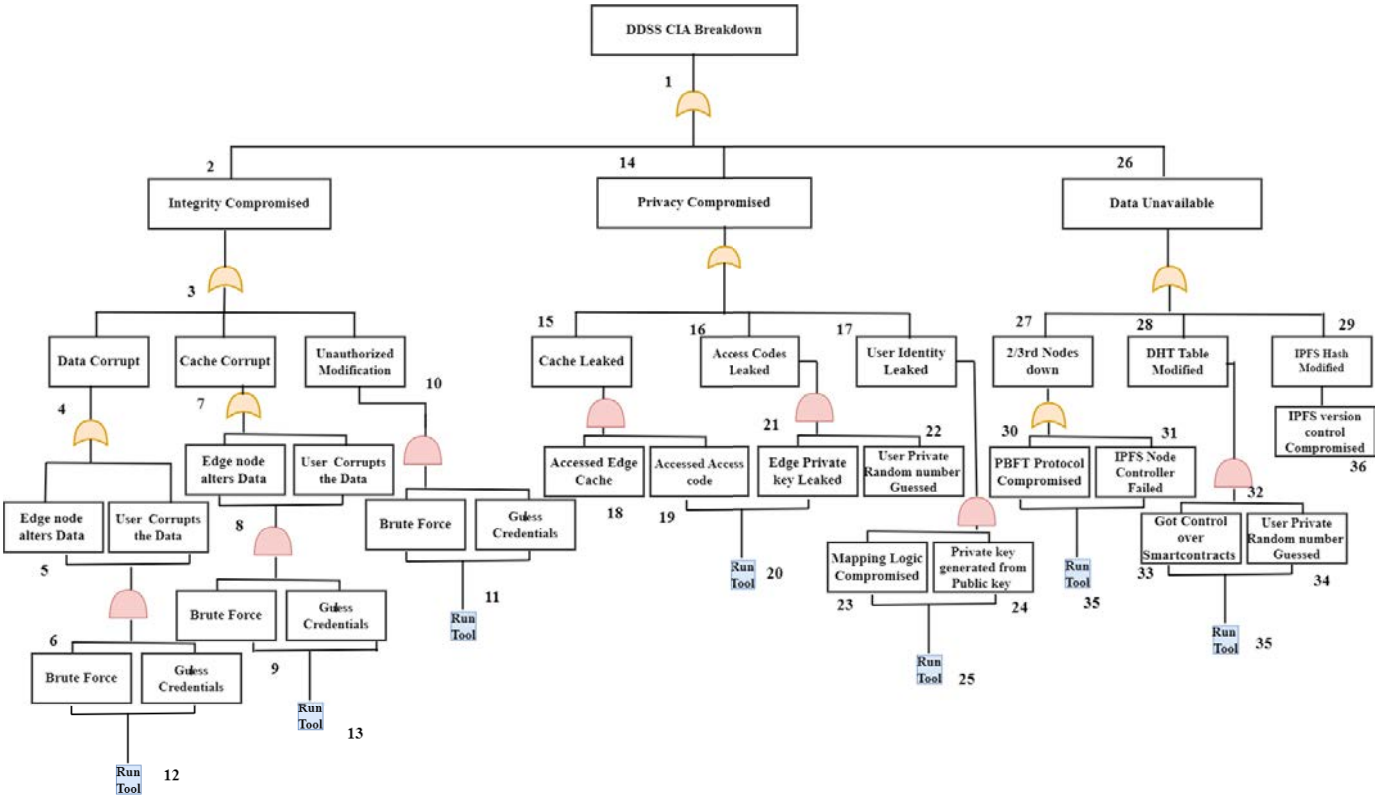


Figure 12: DDSS Internal Security Analysis using Attack Tree

D-AI/ML model improved the system's capabilities for predicting real-time heart attacks with an accuracy of 89.12%. The proposed system outperformed the existing decentralized Fortified-Chain with an approximated value of 17% and six times when compared with the classical cloud-centric H-CPS model in overall system throughput. In the scalability test it is consistently performing 10% improvement over its precede Fortified-Chain. Also, the file operation response latency is almost half of the current Fortified-Chain. Finally, the proposed system provides a platform for different stakeholders in the healthcare industry to make digital agreements using smart contracts. Moreover, the test-bed results showcase that the proposed Fortified-Chain 2.0 can provide decentralized AI/ML services with low latency and high throughput compared to the existing Fortified-Chain. Further, the system analysis revealed that our model could deliver better security and privacy to the medical records on DDSS.

ACKNOWLEDGEMENT

The Authors would like to thank the Science and Engineering Research Board (SERB) for supporting this work, Grant number TAR/2019/000286.

REFERENCES

[1] A. M. Joshi, P Jain, and S. P. Mohanty. iGLU 3.0: A Secure Noninvasive Glucometer and Automatic Insulin Delivery System in IoMT. *IEEE Transactions on Consumer Electronics*, 68(1):14–22, 2022.

[2] L. Rachakonda, A. K. Bapatla, S. P. Mohanty, and E. Kougianos. SaYoPillow: Blockchain-Integrated Privacy-Assured IoMT Framework for Stress Management Considering Sleeping Habits. *IEEE Transactions on Consumer Electronics*, 67(1):20–29, 2021.

[3] Abdul Razaque, Fathi Amsaad, Musbah Abdulgader, Bandar Alotaibi, Fawaz Alsolami, Duisen Gulsezim, Saraju P. Mohanty, and Salim Hariri. A Mobility-Aware Human-Centric Cyber–Physical System for Efficient and Secure Smart Healthcare. *IEEE Internet of Things Journal*, 9(22):22434–22452, 2022.

[4] Y Sun, Frank P.W. Lo, and B Lo. Security and Privacy for the Internet of Medical Things Enabled Healthcare Systems: A Survey. *IEEE Access*, 7:183339–183355, 2019.

[5] S. Biswas, K. Sharif, F. Li, I. Alam, and S. P. Mohanty. DAAC: Digital Asset Access Control in a Unified Blockchain Based E-Health System. *IEEE Transactions on Big Data*, 8(5):1273–1287, 2022.

[6] B. S. Egala, S Priyanka, and A. K Pradhan. SHPI: Smart Healthcare System for Patients in ICU using IoT. In *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6, 2019.

[7] B. S. Egala, A. K. Pradhan, Venkataramana Badarla, and S. P. Mohanty. iBlock: An Intelligent Decentralised Blockchain-based Pandemic Detection and Assisting System. *Journal of Signal Processing Systems*, pages 1–6, 2021.

[8] P Sundaravadivel, E Kougianos, S. P. Mohanty, and M. K Ganapathiraju. Everything you wanted to know about

smart health care: Evaluating the different technologies and components of the Internet of Things for better health. *IEEE Consumer Electronics Magazine*, 7(1):18–28, 2017.

[9] W Shi, J Cao, Q Zhang, Y Li, and L Xu. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[10] M. Rached, Z. Bahroun, and C. Jean-Pierre. Decentralised decision-making with information sharing vs. centralised decision-making in supply chains. *International Journal of Production Research*, 54(24):7274–7295, 2016.

[11] B. S. Egala, A. K. Pradhan, S. Gupta, K S Sahoo, M. Bilal, and K-S Kwak. CoviBlock: A Secure Blockchain-Based Smart Healthcare Assisting System. *Sustainability*, 14(24), 2022.

[12] M. A Akkaş, R SOKULLU, and H E Çetin. Healthcare and Patient Monitoring Using IoT. *Internet of Things*, 11:100–173, 2020.

[13] S. Wang, J. Wang, X. Wang, T. Qiu, Y. Yuan, L. Ouyang, Y. Guo, and F. Wang. Blockchain-powered parallel healthcare systems based on the acp approach. *IEEE Transactions on Computational Social Systems*, 5:942–950, 2018.

[14] A. F. Subahi. Edge-Based IoT Medical Record System: Requirements, Recommendations and Conceptual Design. *IEEE Access*, 7:94150–94159, 2019.

[15] 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*, 6(5):8770–8781, 2019.

[16] R. Ding, H. Zhong, J. Ma, X. Liu, and J. Ning. Lightweight Privacy-Preserving Identity-Based Verifiable IoT-Based Health Storage System. *IEEE Internet of Things Journal*, 6(5):8393–8405, 2019.

[17] E. Daraghmi, Y. Daraghmi, and S. Yuan. MedChain: A Design of Blockchain-Based System for Medical Records Access and Permissions Management. *IEEE Access*, 7:164595–164613, 2019.

[18] A. Alabdulatif, I. Khalil, X. Yi, and M. Guizani. Secure Edge of Things for Smart Healthcare Surveillance Framework. *IEEE Access*, 7:31010–31021, 2019.

[19] Y Yu, S Liu, P L Yeoh, B Vucetic, and Y Li. LayerChain: A Hierarchical Edge-Cloud Blockchain for Large-Scale Low-Delay Industrial Internet of Things Applications. *IEEE Transactions on Industrial Informatics*, 17(7):5077–5086, 2021.

[20] Y Fan, H Wu, and H-Y Paik. DR-BFT: A consensus algorithm for blockchain-based multi-layer data integrity framework in dynamic edge computing system. *Future Generation Computer Systems*, 124:33–48, 2021.

[21] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State. Blockchain-based, decentralized access control for ipfs. In *IEEE International Conference on Internet of Things (iThings)*, pages 1499–1506, July 2018.

[22] A. Awad Abdellatif, L. Samara, A. Mohamed, A. Erbad, C. Fabiana Chiasserini, M. Guizani, M. Dennis O'Connor, and J Laughton. MEdge-Chain: Leveraging Edge Computing and Blockchain for Efficient Medical Data Exchange. *IEEE Internet of Things Journal*, 8(21):15762–15775, 2021.

[23] R. Akkaoui, X. Hei, and W. Cheng. EdgeMediChain: A Hybrid Edge Blockchain-Based Framework for Health Data Exchange. *IEEE Access*, 8:113467–113486, 2020.

[24] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne. BEdgeHealth: A Decentralized Architecture for Edge-Based IoMT Networks Using Blockchain. *IEEE Internet of Things Journal*, 8(14):11743–11757, 2021.

[25] M. A. Khan and F. Algarni. A healthcare monitoring system for the diagnosis of heart disease in the IoMT cloud environment using MSSO-ANFIS. *IEEE Access*, 8:122259–122269, 2020.

[26] W. Chang, Y. Liu, X. Wu, Y. Xiao, and et. al. Zhou. A New Hybrid XGBSVM Model: Application for Hypertensive Heart Disease. *IEEE Access*, 7:175248–175258, 2019.

[27] L. Ali, A. Rahman, and Khan et al. An Automated Diagnostic System for Heart Disease Prediction Based on χ^2 Statistical Model and Optimally Configured Deep Neural Network. *IEEE Access*, 7:34938–34945, 2019.

[28] A. Javeed, S. Zhou, and et. al. Yongjian. An Intelligent Learning System based on Random Search Algorithm and Optimized Random Forest Model for Improved Heart Disease Detection. *IEEE Access*, 7:180235–180243, 2019.

[29] Liaqat Ali, Awais Niamat, and Khan et al. An optimized stacked support vector machines based expert system for the effective prediction of heart failure. *IEEE Access*, 7:54007–54014, 2019.

[30] N. Islam, Y. Faheem, I. U. Din, M. Talha, M. Guizani, and M. Khalil. A blockchain-based fog computing framework for activity recognition as an application to e-Healthcare services. *Future Generation Computer Systems*, 100:569–578, 2019.

[31] P. Bhattacharya, S. Tanwar, U. Bodkhe, S. Tyagi, and N. Kumar. BinDaaS: Blockchain-Based Deep-Learning as-a-Service in Healthcare 4.0 Applications. *IEEE Transactions on Network Science and Engineering*, 8(2):1242–1255, 2021.

[32] T-T. Kuo and L. Ohno-Machado. Modelchain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks. *arXiv preprint arXiv:1802.01746*, 2018.

[33] X. Liu, R. H. Deng, Kim-K. R. Choo, and Y. Yang. Privacy-Preserving Outsourced Clinical Decision Support System in the Cloud. *IEEE Transactions on Services Computing*, 14(1):222–234, 2021.

[34] C. Guo, P. Tian, and Kim-K. R. Choo. Enabling Privacy-Assured Fog-Based Data Aggregation in E-Healthcare Systems. *IEEE Transactions on Industrial Informatics*, 17(3):1948–1957, 2021.

[35] B. S. Egala, A. K. Pradhan, Venkataramana Badarla, and S. P. Mohanty. Fortified-Chain: A Blockchain-Based Framework for Security and Privacy-Assured Internet of Medical Things With Effective Access Control. *IEEE Internet of Things Journal*, 8(14):11717–11731, 2021.

[36] M. A. Salahuddin, A. Al-Fuqaha, M. Guizani, K. Shuaib, and F. Sallabi. Softwarization of Internet of Things Infrastructure for Secure and Smart Healthcare. *Computer*, 50(7):74–79, 2017.

[37] Y. Sungwon and M. Gerla. Personal gateway in mobile health monitoring. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 636–641, March 2011.

[38] L. Breiman. Random Forests. *Machine learning*, 45(1):5–32, 2001.

[39] K. Lauter. The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless Communications*, 11(1):62–67, 2004.

[40] kaggle. Electronic Health Record (EHR) datasets. https://www.kaggle.com/datasets. [Accessed Sep. 2021].

[41] M. Lichman. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2013. [Accessed Sep. 2021].

[42] B. S. Egala and A. K. Pradhan. Fortified-Chain2.0. https://github.com/BhaskaraSanthosh/Fortified-Chain2.0, 2022. [Available Online].

Bhaskara S. Egala received his bachelor's degree in Information Technology from JNTU-K University, in 2011. He has received Post Graduation Diploma in IT Infrastructure, Systems and Security (PG-DITISS) from Centre for Development of Advanced Computing, Pune, 2013. He then commenced his master's in Cyber Security from JNTU-K University, in 2016. Now, he is pursuing Ph.D. degree in SRM University, Amaravati, AP. His current research interest covers Cyber security, Smart Healthcare systems, DHT, Blockchain technology and Metaverse.

Ashok K. Pradhan is currently working as an Associate Professor in the Department of Computer Science & Engineering, School of Engineering and Applied Science at SRM University, Amaravati, AP. He has received his M. Tech degree in the Department of Computer Science and Engineering from the National Institute of Technology (NIT), Rourkela, India, 2010. He has received his Ph.D. degree in the Department of Computer Science and Engineering NIT Durgapur, India, 2015. His areas of interest and research includes Optical Communication and Networks, Internet of Things (IoT), Blockchain Technology, Cyber Security, IoMT, Cloud Computing, Edge Computing, Fog Computing, and Computer Algorithms. He has authored more than 30 research journals/conferences, 2 books and 6 patents published.

Prasenjit Dey is currently working as an Assistant Professor in the Department of Computer Science & Engineering, Cooch Behar Government Engineering College, Cooch Behar,India. He has received the M.Tech. and Ph.D. degrees from the National Institute of Technology Durgapur, India. His areas of interest and research includes machine learning and artificial intelligence, deep learning, and cyber security. He has published more than 10 research papers in reputed peer-reviewed journals/conferences with high impact factors.

Venkata R. Badarla is currently working as an Associate Professor in the Department of Computer Science & Engineering at Indian Institute of Technology, Tirupati, AP. He has received his M.E degree in the Department of Information Systems from the Birla Institute of Technology and Science, Pilani, India, 1997. He has received his Ph.D. degree in the Department of Computer Science and Engineering, from Indian Institute of Technology, Madras, India, 2007. His areas of interest and research includes Wireless Networks, Cloud Computing, and Internet of Things (IoT). He has published more than 30 research papers in reputed peer-reviewed journals/conferences with high impact factors.

Saraju P. Mohanty (Senior Member, IEEE) received the bachelor's degree (Honors) in electrical engineering from the Orissa University of Agriculture and Technology, Bhubaneswar, in 1995, the master's degree in Systems Science and Automation from the Indian Institute of Science, Bengaluru, in 1999, and the Ph.D. degree in Computer Science and Engineering from the University of South Florida, Tampa, in 2003. He is a Professor with the University of North Texas. His research is in "Smart Electronic Systems" which has been funded by National Science Foundations (NSF), Semiconductor Research Corporation (SRC), U.S. Air Force, IUSSTF, and Mission Innovation. He has authored 450 research articles, 5 books, and 9 granted and pending patents. His Google Scholar h-index is 47 and i10-index is 206 with 10,980 citations. He is regarded as a visionary researcher on Smart Cities technology in which his research deals with security and energy aware, and AI/ML-integrated smart components. He introduced the Secure Digital Camera (SDC) in 2004 with built-in security features designed using Hardware Assisted Security (HAS) or Security by Design (SbD) principle. He is widely credited as the designer for the first digital watermarking chip in 2004 and first the low-power digital watermarking chip in 2006. He is a recipient of 16 best paper awards, Fulbright Specialist Award in 2020, IEEE Consumer Electronics Society Outstanding Service Award in 2020, the IEEE-CS-TCVLSI Distinguished Leadership Award in 2018, and the PROSE Award for Best Textbook in Physical Sciences and Mathematics category in 2016. He has delivered 15 keynotes and served on 14 panels at various International Conferences. He has been serving on the editorial board of several peer-reviewed international transactions/journals, including IEEE Transactions on Big Data (TBD), IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), IEEE Transactions on Consumer Electronics (TCE), and ACM Journal on Emerging Technologies in Computing Systems (JETC). He has been the Editor-in-Chief (EiC) of the IEEE Consumer Electronics Magazine (MCE) during 2016-2021. He served as the Chair of Technical Committee on Very Large Scale Integration (TCVLSI), IEEE Computer Society (IEEE-CS) during 2014-2018 and on the Board of Governors of the IEEE Consumer Electronics Society during 2019-2021. He serves on the steering, organizing, and program committees of several international conferences. He is the steering committee chair/vice-chair for the IEEE International Symposium on Smart Electronic Systems (IEEE-iSES), the IEEE-CS Symposium on VLSI (ISVLSI), and the OITS International Conference on Information Technology (OCIT). He has mentored 2 post-doctoral researchers, and supervised 14 Ph.D. dissertations, 26 M.S. theses, and 18 undergraduate projects.