Private Blockchain-Based AI-Envisioned Home Monitoring Framework in IoMT-Enabled COVID-19 Environment

Basudeb Bera, Ankush Mitra, and Ashok Kumar Das
International Institute of Information Technology

Deepak Puthal
Newcastle University

YoungHo Park
Kyungpook National University

***Abstract*—Coronavirus disease-2019 (COVID-19) is a very serious health concern to the human life throughout the world. The Internet of Medical Things (IoMT) allows us to deploy several wearable Internet of Things-enabled smart devices in a patient's body. The deployed smart devices should then securely communicate to nearby mobile devices installed in a smart home, which then securely communicate with the associated fog server for information processing. The processed information in terms of transactions are formed as blocks and put into a private blockchain consisting of cloud servers. Since the patient's vital signs are very confidential and private, we apply the private blockchain. This article makes utilization of fog computing and blockchain technology simultaneously to come up with more secure system in an IoMT-enabled COVID-19 situation for patients' home**

Digital Object Identifier 10.1109/MCE.2021.3137104
Date of publication 21 December 2021; date of current version 8 April 2023.

monitoring purpose. We first discuss various phases related to development of a new fog-based private blockchain-enabled home monitoring framework. Next, we discuss how artificial intelligence-enabled big data analytics helps in analyzing and tracking the patients' information related to COVID-19 cases. Finally, a blockchain implementation has been performed to exhibit practical demonstration of the proposed blockchain system.

To track a patient in a home monitoring system, specifically during the coronavirus disease-2019 (COVID-19) pandemic, the Internet of Medical Things (IoMT) plays a very important role.[1–3] In IoMT environment, a person's body is deployed with various wearable/implantable smart devices, which communicate with the nearby mobile device(s) owned by that person. The information collected by the mobile device is then stored in the cloud server (CSs).

Fog computing is a distributed computing application consisting of a number of servers that can perform computation and networking, and provide storage similar to the servers in a cloud data center. Fog computing essentially aims to bring server resources closer to the devices involved in the generation of data. It increases the intelligence of local area network by allowing computation of the data to be performed using the resource capabilities available inside the network where the data gathering devices exist. This helps to reduce latency in response times that are encountered in cloud computing where the data was needed to be transmitted to servers placed in different geographical locations before any processing could begin. Fog computing has also allowed increased security of data by allowing highly sensitive data to be processed at fog servers and only low-sensitive data to be forwarded to the CS. It also promotes better management of huge volumes of data by distributing the data among multiple nodes in the local network.

Fog computing can be used in conjunction with cloud computing and edge computing. Cloud computing consists of multiple high-resource servers placed inside a data center owned by a service provider. Any user needing resources will associate with the provider and pay for the amount of resource used, without the need for delving into the details of managing these resources. On the other hand, fog computing allows the processing to be performed at the local network of the data gathering device, whereas edge computing allows the processing to be performed at either the devices that hold the sensors or a gateway node placed in close physical proximity to these sensor devices.

Internet of Things (IoT) is a collection of highly diverse devices with the ability to read the physical parameters of their surroundings, process the data in a distributed manner, and perform collective actions based on the processed data, using the Internet and with minimal human intervention. Fog computing has major applications in the IoT world, where a huge amount of data sensed by the IoT smart sensor devices are regularly sent to the CSs for processing. The working of IoT is highly reliant on real-time processing of sensor data as the user is in continuous interaction with the smart devices. In such a scenario, latency due to processing and network transmission may be highly deterrent to the smooth functioning of many IoT applications. In addition, data in IoT applications are sensitive to the user and requires protection from any unprecedented misuses. This shows that fog computing is supremely relevant in the context of IoT applications.

The distributed nature of IoT and fog computing mandates that the security provided to the resources of networking and data on these applications is also based on a distributed structure. The nodes in a fog computing environment should be working with equal capacities without any necessity of trust among them, in order to allow the layers and infrastructure that constitute the stack of a fog node to be owned and managed by different entities. This idea of security necessitates the need for distributed trust in fog computing that can be realized using the concept of blockchain technology. Blockchain is a distributed and decentralized system that stores any data in a ledger in the form of transactions inside a block and connects multiple such blocks together as a chain. The distributed nature of a blockchain is in the replication of the blockchain among the nodes in a network using mining strategies and the collective maintenance of the blockchain for consistency in

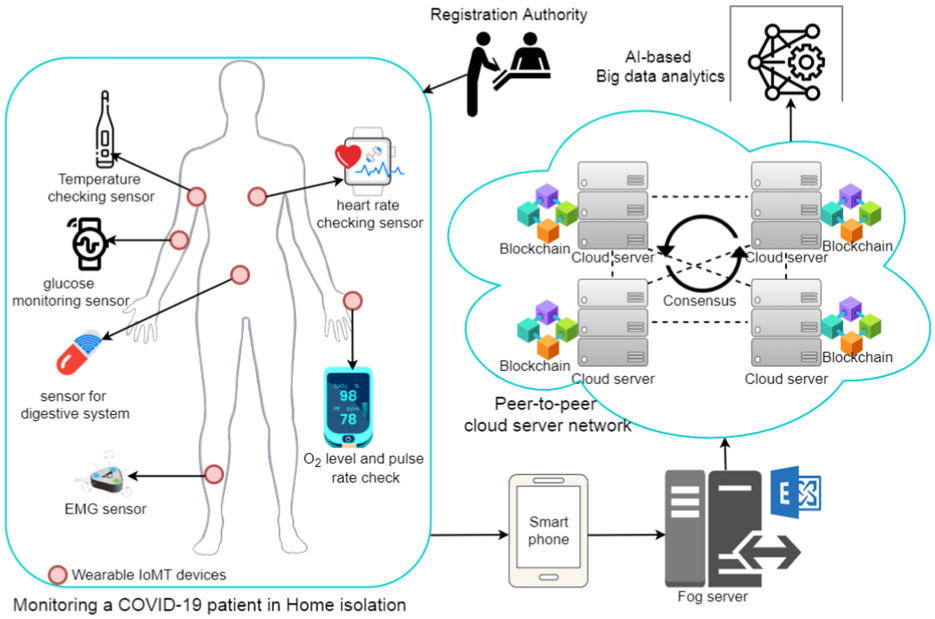


FIGURE 1. Generic IoMT-based blockchain-enabled home monitoring framework.

the replicated data in the blocks among the nodes using consensus strategies. The stored data are persistent allowing auditability, but it also ensures anonymity and untraceability.

Blockchain is compatible with fog computing as it allows the devices nodes to be used as blockchain nodes and fog nodes to be used as the miner nodes. The idle resources available with the nodes in a fog network can be used for blockchain maintenance. The nodes that allow their idle resources to used for blockchain processing, can be rewarded in accordance with the amount of resources provided, by using an appropriate consensus mechanism. In recent years, the researchers have discussed and designed security protocols in fog- and cloud-based IoMT environment using the blockchain technology.[4–7] In addition, it was also shown that the blockchain technology is much viable solution to tackle COVID-19 situation for patients' tracking purpose.[8–11]

SYSTEM MODEL

In the considered system model shown in Figure 1, we first consider the patients whose body is equipped with several smart wearable devices and these devices are connected to each other to form wireless body area network. These wearable devices have the capability to sense vital signs from the patient's body and the collected information (data) are securely sent to its owned mobile device(s). The mobile devices then send securely data to the attached fog sever. The in-charge fog server filters the information and the useful information is then converted to the transactions. The transactions are then encrypted using the public key of the fog server. Each encrypted transaction is then used to produce a digital signature using the elliptic curve digital signature algorithm, say ECDSA.Sig (.). Once the encrypted transactions and their signatures are received by the CSs, the blocks are formed. The created blocks are then mined in order to validate and add into the private blockchain. In addition, we have also artificial intelligence (AI)-based big data analytics module that is responsible for analyzing the information stored in the blocks and making correct predictions on the datasets.

PROPOSED BLOCKCHAIN FRAMEWORK FOR HOME MONITORING

In this section, we elaborate the proposed framework that considers blockchain technology followed by the AI-based big data analytics.

It is assumed that the patients and their wearable devices and mobile devices, and fog and CSs are registered by a trusted registration authority prior to their deployment/installment

for home monitoring purpose. Furthermore, it is assumed that the wearable devices will securely communicate with the mobile device of a patient by establishing session keys among them. In addition, the mobile device and fog server will also establish secret key among them for their secure communication. On the other hand, secure communication among the fog server and the associated CSs are done using the elliptic curve cryptography-based encryption and decryption.

Block creation, verification as well as addition, are done in a peer-to-peer (P2P) CS network, called P2P CSN. Once the data are securely received from the fog server(s) in the form of transactions by the peer nodes, the transaction is updated into each node's transactions pool. This means that each peer node has a transactions pool that is upgraded by the new one. A new block is created by a proposer (also called a leader), which is elected by a round-robin fashion from the P2P CSN once the transactions pool reaches to a certain predefined transaction threshold value. The block addition and verification are executed by a widely adopted distributed algorithm, named as "Practical Byzantine Fault Tolerance algorithm."[12]

The entire process of the voting-based consensus algorithm is described as follows.

- The proposer broadcasts the proposed block into the entire network. Other nodes (called the followers) receive the proposed block that contains some transactions.
- After that, the followers verify the block with their maintained own transactions pool.
- If the injected transactions in the proposed block are successfully verified with their existing pool, then the block is called verified.
- Finally, the block will be added to the blockchain.

The general structure of a block in the private blockchain is shown in Figure 2. It consists of block header and block payload as follows.

Block Header: It has the following fields.

- *Block Version:* It is a unique number that represents a block in the blockchain.
- *Previous Block Hash:* It denotes the hash value of the contents of the previous block of a current block to be added into the blockchain.

Block Header
Block Version
Previous Block Hash (PBH)
Merkle Tree Root (MTR)
Timestamp
Signer's Public Key
Block Payload (Transactions)
(Encrypted Transactions (ETx), ECDSA.Sig(ETx))
Current Block Hash (CBH)

FIGURE 2. Private-block structure.

- *Merkle Tree Root:* It is the hash value of all the transactions present in a block, which is used to verify the transactions of that block.
- *Timestamp:* It is the time when a block was created.
- *Signer's Public Key:* It is the public key that is used to verify the signature of an encrypted transaction ETx, ECDSA.Sig(ETx), using the elliptic curve digital signature verification algorithm, ECDSA.Ver(.), that is, $\{0, 1\} \leftarrow$ ECDSA.Ver[ECDSA.Sig(ETx)].

Block Payload: It contains a specific number of encrypted transactions and their signatures pairs of the form: $(\mathrm{ETx}$, ECDSA.Sig$(\mathrm{ETx}))$.

Current Block Hash: It is finally the hash value of all the fields contained in the block header and block payload.

Combination of blockchain and AI can help in the big data analysis process on the information stored on blocks in a private blockchain for better prediction of results and tracking of patients who are under the home monitoring system in the proposed framework. There are several attacks that can be performed by an attacker, such as "Poisson noise insertion attack" and "data poisoning attack" on the data that are simply stored in the CS. Since the blockchain provides immutability, transparency, and decentralization, once the blocks are validated successfully, the information stored in the blocks is treated as genuine. Hence, by applying the AI and machine learning (ML)-based approaches on the authenticated datasets stored in the private blockchain, correct predictions are quite expected. As a result, better big data analytics for the patient's COVID-19 data including patient tracking is viable.

The overall process containing in the proposed framework for COVID-19-related patients

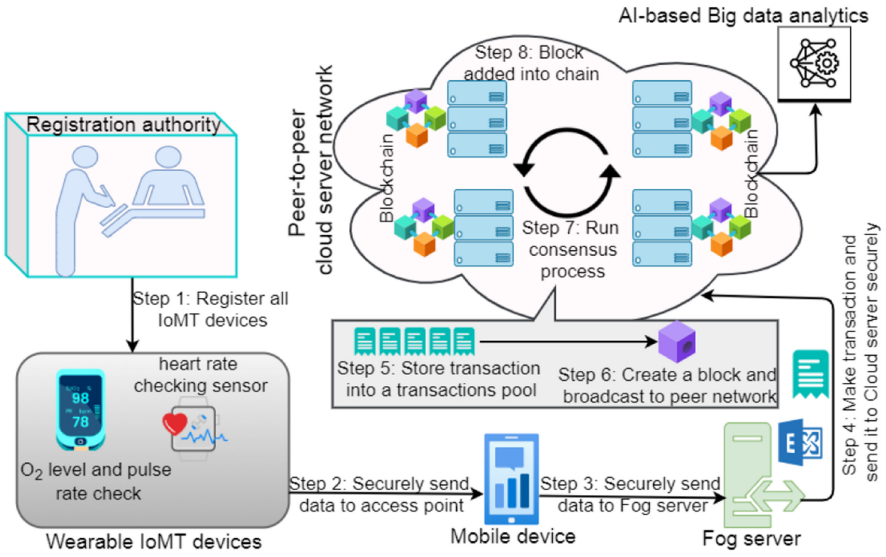


FIGURE 3. Overall process in the proposed framework.

tracking under a smart home environment is then demonstrated in Figure 3.

BLOCKCHAIN IMPLEMENTATION

The real-time blockchain simulation has been executed over a system configuration considered as a CS setting with the environment: "Ubuntu 20.04 LTS, Intel Core i5-8400 CPU 3855 U @ 1.60 GHz, Memory 4 GB, OS type 64-bit, disk size 152.6 GB."

We set up a distributed CS system for blockchain execution, which forms a P2P CS network. The P2P CS network consists of the number of distributed servers as 12, which are participated for blockchain simulation purpose. The communication between the peer nodes are made by the message-passing process, where the messages may be transactions or a block. The script for this simulation is written in node.js language with VS CODE 2019. In this simulation, we consider the genesis block (first block in the blockchain) to have all peer nodes before adding new block into their database. We have then implemented the voting-based "PBFT consensus algorithm" discussed previously for block verification and addition into the blockchain. The simulation results are based on the number of blocks mined versus computational time (in seconds) (the time required to add the block into blockchain) and transactions with computational time. The details of these two cases are elaborated as follows.

- *Case I*: In this case, we consider the number of blockchain versus the computational time (in seconds), where "a fixed number of transactions for each block in the blockchain" is taken as 34. This means the chain of block is varied but the loaded transaction in each block in varied chain is fixed and the simulation outcomes are reported in Figure 4(a). The results imply whenever the blocks are increased, the computational time also increases in proportion. The graph of this result is a straight line, which implies a linear type. Therefore, "when the number of blocks mined is increased, the total computational time increases linearly."
- *Case II*: In this case, we have presented the simulation results in Figure 4(b), where we have considered a fixed number of blocks to be mined in each blockchain as 40, and only varied the number of transactions. It is observed that the outcome trends show that the total computational time (in seconds) increases linearly (based on the linear type graph, that is, the graph represents a single straight line) when the number of transactions per block also increases.

EXPERIMENTAL RESULTS FOR AI-ENABLED BIG DATA ANALYTICS

In this section, we provide the comprehensive experimental results under two circumstances:

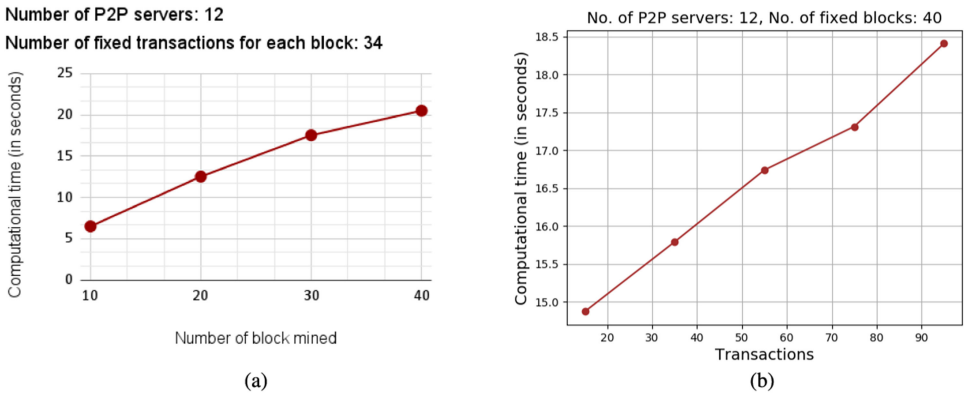


FIGURE 4. Blockchain simulation result for various cases. (a) Blockchain simulation result for Case I. (b) Blockchain simulation result for Case II.

1) performance of ML model under data poisoning attacks and 2) performance of ML model without data poisoning attacks. In the first case, we show how the data poisoning attacks can reflect the ML model when the data are on some cloud storage (i.e., not in the blockchains), whereas in the second case, we show the effect when the data are in the blockchains (i.e., without data poisoning attacks).

Performance Metrics

To depict the correct performance analysis of the ML models under data poisoning attack, we have used four important ML metrics. Assume that TP_n, TN_n, FP_n, and FN_n denote the "true positive," "true negative," "false positive," and "false negative," respectively, for the nth experiment. The following metrics are then defined.

Accuracy Accuracy score is a very important performance metric, which can be used to depict the correct prediction score of the ML models. The accuracy can be defined mathematically as $\mathrm{Accuracy} = \frac{\mathrm{TP}_n+\mathrm{TN}_n}{\mathrm{TP}_n+\mathrm{TN}_n+\mathrm{FP}_n+\mathrm{FN}_n}$.

Recall Recall is one of the most useful performance metrics that can be used in ML to show the true positive rate of the model. It can be defined as $\mathrm{Recall} = \frac{\mathrm{TP}_n}{\mathrm{TP}_n+\mathrm{FN}_n}$.

Precision Precision is another very important performance metric, which depicts the precision of the ML model. It used to compare the number of true positives that a model claim with the total true positives in the dataset. It is defined as $\mathrm{Precision} = \frac{\mathrm{TP}_n}{\mathrm{TP}_n+\mathrm{FP}_n}$.

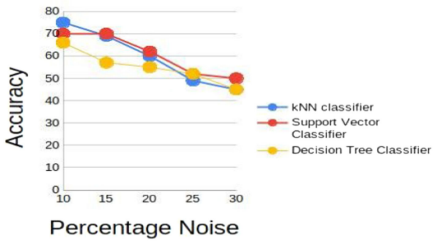
F1 Score F1 score is most important metric used in ML to show how the ML model trades with both recall and precision. It is the weighted average of recall and precision of the ML model. It is defined as $\mathrm{F1\ Score} = \frac{2*\mathrm{TP}_n}{2*\mathrm{TP}_n+\mathrm{FP}_n+\mathrm{FN}_n}$.

Performance of ML Model Under Data Poisoning Attacks

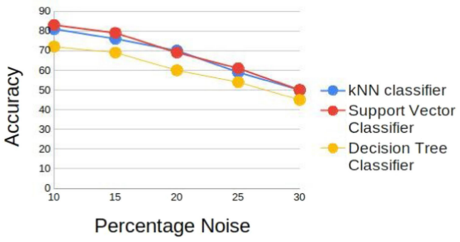
This section describes about the performance of the ML model under data poisoning attacks.

Correct data are crucial for the ML model to perform better. If the training data are corrupted by any means, they can hamper the overall performance of the ML model. Data that are present in any centralized database can easily be corrupted as the database does not ensure the tamper proof ability of the data storage. In contrast, if the data are present in any blockchain system, they are guaranteed to be tamper proof by the design of the blockchain system. So, if the data are not stored in any blockchain system, various data poisoning attacks are possible on the training data of the ML model. In the following, we discuss about two types of data poisoning attacks, namely 1) salt noise injecting attack and 2) label flipping attack.

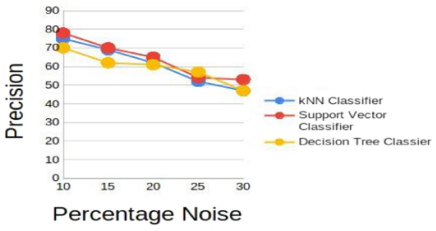
- *Salt noise injecting attack:* In the case of the salt noise injecting attack, an attacker gains the access of the data storage and then uses the standard salt noise function to corrupt the data in a meaningful way.
- *Label flipping attack:* In the case of the label flipping attack, the attacker first gains the



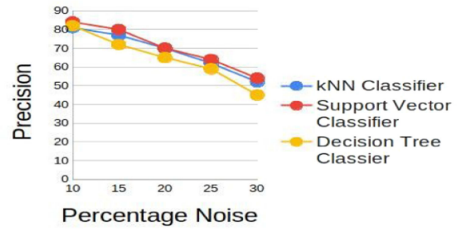
(i) Effect of accuracy under label flipping attacks



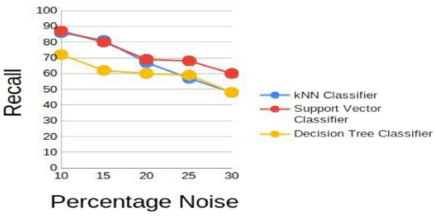
(i) Effect of accuracy under salt noise attacks



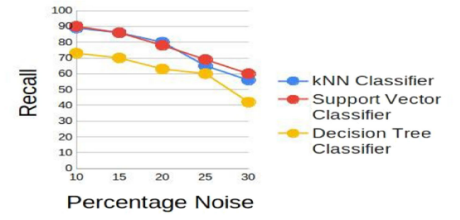
(ii) Effect of precision under label flipping attacks



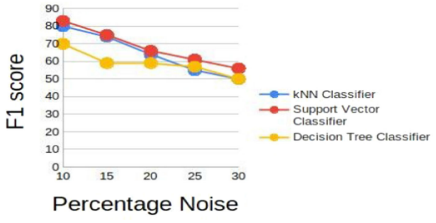
(ii) Effect of precision under salt noise attacks



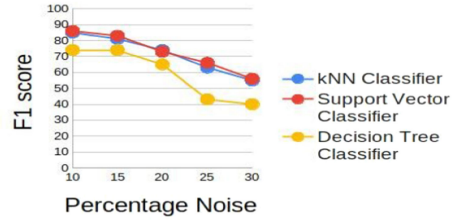
(iii) Effect of recall under label flipping attacks



(iii) Effect of recall under salt noise attacks



(iv) Effect of F1 score under label flipping attacks



(iv) Effect of F1 score under salt noise attacks

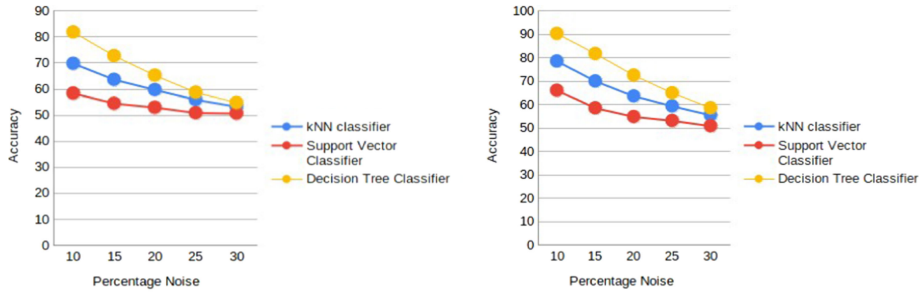
(a) (b)

FIGURE 5. Experimental results under label flipping and salt noise attacks in Case I. (a) Experimental results under label flipping. Effect of (i) accuracy, (ii) precision, (iii) recall, and (iv) F1 score under label flipping attacks. (b) Experimental results under salt noise attacks attacks. Effect of (i) accuracy, (ii) precision, (iii) recall, and (iv) F1 score under salt noise attacks.

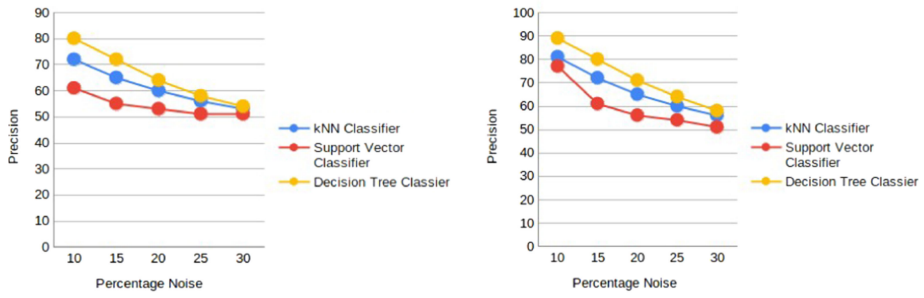
access of the training dataset, and then chooses some (or all) data randomly and modify the ground truth label of those data to corrupt the training dataset in a meaningful way.

To simulate the effect of the data poisoning attack, we have used three different popularly known ML algorithms: a) k-nearest neighbors algorithm (kNN) classifier, b) support vector classifier, and c) decision tree classifier. For each round of experiments, we have recorded the accuracy, precision, recall, and F1 score.

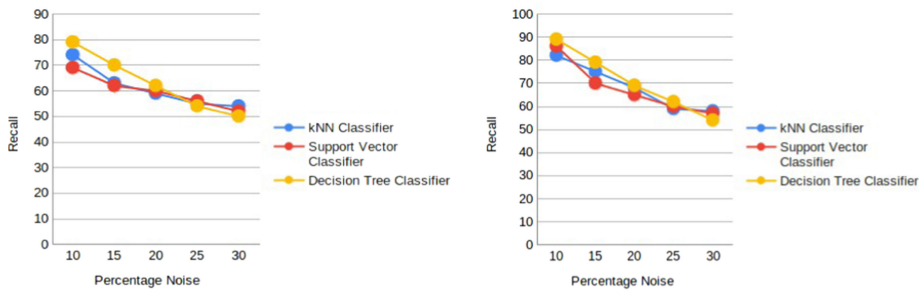
1) Datasets In the following, we consider the following two types of datasets that are used for the experimental purposes.



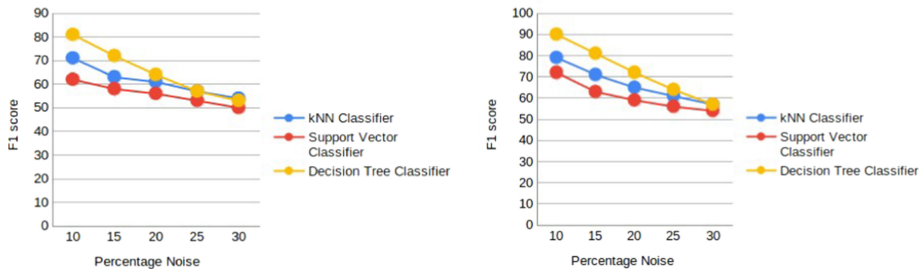
(i) Effect of accuracy under label flipping attacks (i) Effect of accuracy under salt noise attacks



(ii) Effect of precision under label flipping attacks (ii) Effect of precision under salt noise attacks



(iii) Effect of recall under label flipping attacks (iii) Effect of recall under salt noise attacks



(iv) Effect of F1 score under label flipping attacks (iv) Effect of F1 score under salt noise attacks

(a) (b)

FIGURE 6. Experimental results under label flipping and salt noise attacks in Case II. (a) Experimental results under label flipping. Effect of (i) accuracy, (ii) precision, (iii) recall, and (iv) F1 score under label flipping attacks. (b) Experimental results under salt noise attacks attacks. Effect of (i) accuracy, (ii) precision, (iii) recall, and (iv) F1 score under salt noise attacks.

- *Case I*: In this case, we have considered "Heart Disease UCI" dataset[13] that is related to non-COVID-19 related dataset. This dataset has several fields, such as "age (age in years)," "sex (1 = male; 0 = female)," "cp (chest pain type)," "trestbps (resting blood pressure (in mm Hg on admission to the hospital))," "chol (serum cholesterol in mg/dl)," "fbs (fasting blood sugar; 120 mg/dl) (1 = true; 0 = false)," "restecg (resting electrocardiographic results)," "thalach (maximum heart rate achieved)," "exang

(exercise induced angina (1 = yes; 0 = no)),” and “oldpeak (ST depression induced by exercise relative to rest)”.

- *Case II*: In this case, we have taken the *Open Data*, the “General Directorate of Epidemiology,”[14] which is related to COVID-19 dataset. The dataset contains various important attributes, like “sex” “patient type,” “entry date,” “date symptoms,” “date died,” “pneumonia,” “age,” “pregnancy,” “icu,” etc. This dataset was also used to predict if a patient needs to be shifted to ICU based on the patent attribute.

2) Discussion on Experimental Results

We have plotted the values in the graphs as shown in Figures 5 and 6 for the Cases I and II datasets, respectively. It is worth noticing that when the noise level is increased by an attacker, the performance degrades significantly in both the cases.

Performance of ML Model Without Data Poisoning Attacks

It is noticed that if the data are stored in blockchain system, they are by default tamper proof and no one can tamper the data, but can access the data without modifying it with proper credential to decrypt the transactions into the blocks. So, if the training data are present on a blockchain system, the possibility of data poisoning attacks by an attacker is reduced significantly.

Table 1 describes the performance of the ML models without data poisoning attacks. It is assumed that 0% noise insertion in the datasets means no data poisoning attacks. Compared with the results shown in Figures 5 and 6, it is clear that if the data are put into private blockchains, the performance of the ML models significantly improved (see Table 1).

TABLE 1. Performance of ML models without data poisoning attacks.

ML Classifier	Accuracy	Recall	Precision	F1 Score
kNN (Case I)	0.81	0.82	0.88	0.85
kNN (Case II)	0.78	0.82	0.76	0.79
SVM (Case I)	0.85	0.86	0.91	0.87
SVM (Case II)	0.66	0.85	0.62	0.71
Decision Tree (Case I)	0.73	0.78	0.73	0.75
Decision Tree (Case II)	0.91	0.89	0.91	0.90

CONCLUSION

In this article, a private blockchain-enabled security system has been suggested for home monitoring in the IoMT-enabled COVID-19 situation. The proposed system makes use of both blockchain technology and AI-based big data analytics. The blockchain-based implementation has been performed in computing computational time when the number of blocks mined and the number of transactions in a block are varied. It is also shown that the blockchain helps in achieving better security and at the same time, AI also helps for efficient big data analytics. In recent years, the lattice-based cryptographic techniques have been applied in healthcare systems,[15] because of its superior security and computational efficiency as compared to the traditional public-key cryptosystems. In future, we would like to improve the proposed framework using lattice-based cryptosystem.

ACKNOWLEDGMENTS

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1I1A3058605.

REFERENCES

1. R. Guo, G. Yang, H. Shi, Y. Zhang, and D. Zheng, “O^3-R-CP-ABE: An efficient and revocable attribute-based encryption scheme in the cloud-assisted IoMT system,” *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8949–8963, Jun. 2021, doi: 10.1109/JIOT.2021.3055541.
2. A. K. Das, M. Wazid, N. Kumar, M. K. Khan, K. K. R. Choo, and Y. Park, “Design of secure and lightweight authentication protocol for wearable devices environment,” *IEEE J. Biomed. Health Informat.*, vol. 22, no. 4, pp. 1310–1322, Jul. 2018, doi: 10.1109/JBHI.2017.2753464.
3. N. Garg, M. Wazid, A. K. Das, D. P. Singh, J. J. P. C. Rodrigues, and Y. Park, “BAKMP-IoMT: Design of blockchain enabled authenticated key management protocol for Internet of Medical Things deployment,” *IEEE Access*, vol. 8, pp. 95956–95977, 2020, doi: 10.1109/ACCESS.2020.2995917.
4. M. Seliem and K. Elgazzar, “BIoMT: Blockchain for the Internet of Medical Things,” in *Proc. IEEE Int. Black Sea Conf. Commun. Netw.*, 2019, pp. 1–4, doi: 10.1109/BlackSeaCom.2019.8812784.

5. D. C. Nguyen, K. D. Nguyen, and P. N. Pathirana, "A mobile cloud based IoMT framework for automated health assessment and management," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2019, pp. 6517–6520, doi: 10.1109/EMBC.2019.8856631.
6. B. S. Egala, A. K. Pradhan, V. R. 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 Things J.*, vol. 8, no. 14, pp. 11717–11731, Jul. 2021, doi: 10.1109/JIOT.2021.3058946.
7. D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "BEdgeHealth: A decentralized architecture for edge-based IoMT networks using blockchain," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11743–11757, Jul. 2021, doi: 10.1109/502 JIOT.2021.3058953.
8. V. Chamola, V. Hassija, V. Gupta, and M. Guizani, "A comprehensive review of the COVID-19 pandemic and the role of IoT, drones, AI, blockchain, and 5G in managing its impact," *IEEE Access*, vol. 8, pp. 90225–90265, 2020, doi: 10.1109/ACCESS.2020.2992341.
9. S. Ahir, D. Telavane, and R. Thomas, "The impact of artificial intelligence, blockchain, big data and evolving technologies in coronavirus disease—2019 (COVID-19) curtailment," in *Proc. Int. Conf. Smart Electron. Commun.*, 2020, pp. 113–120, doi: 10.1109/ICOSEC49089.2020.9215294.
10. P. K. Singh, S. Nandi, K. Ghafoor, U. Ghosh, and D. B. Rawat, "Preventing COVID-19 spread using information and communication technology," *IEEE Consum. Electron. Mag.*, vol. 10, no. 4, pp. 18–27, Jul. 2021, doi: 10.1109/MCE.2020.3047608.
11. M. S. Hossain, G. Muhammad, and N. Guizani, "Explainable AI and mass surveillance system-based healthcare framework to combat COVID-I9 like pandemics," *IEEE Netw.*, vol. 34, no. 4, pp. 126–132, Jul./Aug. 2020, doi: 10.1109/MNET.011.2000458.
12. M. Castro and B. Liskov, "Practical Byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, 2002, doi: 10.1145/571637.571640.
13. Heart disease UCI, 2018. Accessed: Sep. 2021. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Heart+Disease
14. Open data general directorate of epidemiology. Accessed: Oct. 2021. [Online]. Available: https://www.gob.mx/salud/documentos/datos-abiertos-152127
15. R. Chaudhary, A. Jindal, G. S. Aujla, N. Kumar, A. K. Das, and N. Saxena, "LSCSH: Lattice-based secure cryptosystem for smart healthcare in smart cities environment," *IEEE Commun. Magazine*, vol. 56, no. 4, pp. 24–32, Apr. 2018, doi: 10.1109/MCOM.2018.1700787.

Basudeb Bera is a currently a Ph.D. student in computer science and engineering with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. Contact him at basudeb.bera@research.iiit.ac.in.

Ankush Mitra received the M.Tech. degree in computer science and information security from the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. Contact him at ankush.mitra@students.iiit.ac.in.

Ashok Kumar Das is currently an associate professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. He is a senior member of the IEEE. He is the corresponding author of this article. Contact him at iitkgp.akdas@gmail.com.

Deepak Puthal currently an assistant professor with the School of Computing, Newcastle University, Newcastle upon Tyne, U.K. Contact him at deepak.puthal@newcastle.ac.uk.

YoungHo Park is currently a professor with the School of Electronics Engineering, Kyungpook National University, Daegu, Republic of Korea. He is a member of the IEEE. He is the corresponding author of this article. Contact him at parkyh@knu.ac.kr.