Blockchain-Enabled Online Diagnostic Platform of Suspected Patients of COVID-19 Like Pandemics

Mahmoud Abouyoussef, Surbhi Bhatia, Poonam Chaudhary, Saurabh Sharma, and Muhammad Ismail

Abstract

During times of pandemics, the healthcare system may collapse due to the high demand for healthcare resources. Hence, there is a need for an online-automated platform that enables remote collection of symptoms from suspected patients, accurate and fast diagnostics, and data sharing among different entities within the healthcare system. However, many privacy and scalability challenges face such a platform. To address such challenges, we propose a custom-designed blockchain enabled platform that guarantees privacy-preservation via a mixture of group signature and random numbers that support anonymity of suspected patients and unlinkability of data while enabling mutual interaction between the suspected patient and the platform; provides automatic diagnostics via a deep neural network-based detector that runs on a smart contract within the blockchain; and offers access and administrative authority of the healthcare entities to the database of symptoms and their diagnoses via a consortium-based blockchain architecture. Experimental studies demonstrate a detection accuracy of 90 percent based on a deep convolutional recurrent neural network. A case study of 500 expected patients is examined giving promising results. Every patient can know the test results after only 14 min of submitting the data. The storage requirements are as low as 0.52 MB for each suspected patient and 0.6 MB for each hospital.

Introduction

While the COVID-19 pandemic may feel like a rare event, scientific reports indicate that the pace of pandemics is dramatically accelerating with a rate of nearly one infectious disease emerging every eight months [1]. The world experience with COVID-19 has made it clear that pandemics pose a risk to both humans' health and the global economy. The lessons learned from COVID-19 helped in developing effective strategies to better prepare for the next pandemic.

General Requirements and Challenges

One valuable lesson learned from the COVID-19 pandemic is that the very high demand for resources may result in further spread of the disease and the collapse of the healthcare system.

Unfortunately, the healthcare system functions as a collection of entities, e.g., hospitals, medical and research centers, etc., that operate in isolation. Hence, there is a need for more data sharing among such entities to better learn about the symptoms, any new evolutions, and the relevant test results. In addition, this data sharing will help in enhancing the diagnostic test and improving its accuracy.

Consequently, there is a need for an online platform that can assist in remote collection of indicative symptoms from the suspected patients; virtually running a relevant diagnostic test; online sharing of test results with the suspected patient; and online sharing of the symptoms and their corresponding test results with the healthcare entities. The Internet-of-Medical-Things (IoMT) paradigm empowers these features. However, this platform faces the following challenges. First, the data collection and result reporting phases should not reveal the identity of the suspected patient. Such data and results represent confidential information, and hence, must be protected. The challenge herein is how to hide the identity of the suspected patient and at the same time enable efficient interaction between the platform and the suspected patient for data submission and result reporting. In addition, while being privacy-preserving, there is a need for informing the hospital or medical center of any reported positive case to provide the necessary treatment on time. With such a requirement in mind, the diagnostic test should run autonomously and provide fast results to the suspected patients. Furthermore, the platform should allow continuous updates of the diagnostic test to enhance the reported results as we learn more about the infectious disease. Finally, there is a need to enforce access control where: suspected patients have access only to the diagnostic test results, and healthcare entities should have access to both symptoms and test results of anonymous patients and suspected patients.

Mahmoud Abouyoussef and Muhammad Ismail are with Tennessee Tech University, USA.

Surbhi Bhatia is with King Faisal University, Saudi Arabia.

Poonam Chaudhary is with MRIIRS Faridabad, India.

Saurabh Sharma is with NorthCap University, India.

Digital Object Identifier: 10.1109/IOTM.1001.2100046

Blockchain Motivation

Enabling the suspected patients to send their data to the diagnosing entity and receive the results requires privacy-preserving two-way communication support (for data submission and result reporting). Dedicated point-to-point messaging could not be used as it reveals the identity of the suspected patient. In addition, using a third party to pass the messages from the suspected patient to the diagnosing entity and vice versa is not recommended as this third party is a single point of failure. In this application, one entity (i.e., the diagnosing entity) is known and the suspected patients are the ones who need to protect their privacy. Hence, an anonymous message could be sent from the suspected patients to the diagnosing entity, including the symptoms and an untraceable pointer. The diagnosing entity cannot reply by a dedicated message to the suspected patient because the suspected patient is unknown to it. Accordingly, it broadcasts all the replies for different suspected patients on a distributed ledger. There are various distributed ledger

 2576-3180/21/$25.00 © 2021 IEEE

technologies, but blockchain technology is the best to use due to its attractive features such as immutability, transparency, and the use of smart contracts. These smart contracts help in designing a completely autonomous system for testing and data sharing. Moreover, the smart contracts ensure that all the healthcare entities agree on the same diagnostic test whose rules cannot change except after a predetermined consensus between all the healthcare entities.

Blockchain-Based Platform

Blockchain is employed in the literature to help in issuing medical passports and immunity certificates [2], contact tracing [3], data sharing among medical centers [4], and developing a diagnostic model based on federated learning [5]. However, none of the reported solutions present a comprehensive platform for remote data collection, sharing, and diagnostic testing. The following limitations are observed, despite the previously mentioned advancement in the field. First, most of the existing solutions exploit the traditional blockchain anonymity feature supported by the pair of public-private keys provided to the users [2, 4]. However, this solution does not present complete privacy-preservation as all the data submitted by the suspected patient can be linked to its key. Hence, while anonymity is guaranteed, data unlinkability is absent. Second, to enhance privacy-preservation while running diagnostic tests, differential privacy through noise addition has been adopted in [6]. However, differential privacy via noise addition presents a trade-off between detection accuracy and privacy-preservation level. For instance, the reported test accuracy in [6] ranges between 50 and 86 percent, depending on the desired privacy level.

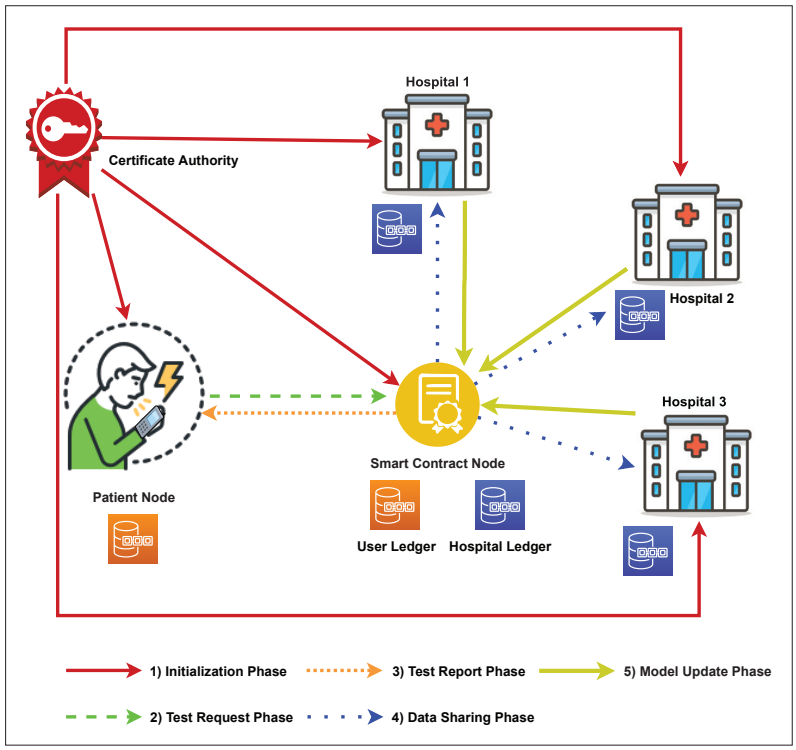

FIGURE 1. Illustration of the system architecture.

Contributions

To present an online platform for remote data collection, sharing, and diagnostic testing, the following contributions are carried out:

- A consortium blockchain networking strategy that integrates different entities of the healthcare system including hospitals and medical and research centers is introduced. Such a model can be enforced by governments as an emergency action when facing pandemics. The proposed platform offers the following attractive features:
  - –The platform requires the collection of simple physiological signals such as coughing sounds in order to develop a database (ledger) of symptoms of suspected patients. Hence, the data can be collected by suspected patients at home and submitted online for diagnostic testing.
  - –The platform supports privacy-preservation through a mixture of group signature and random numbers that enable anonymity of the suspected patient's identity and unlinkability of the data submitted by the same suspected patient at different testing queries while providing the necessary interaction between the suspected patient and the platform for data collection and reporting of test results.
  - –The group signature allows the identification of a suspected patient that is detected to be a positive case for timely treatment.
- The platform relies on a novel random generation mechanism that operates in a distributed manner at the suspected patient's device while ensuring uniqueness over users and time.
- The platform adopts a smart contract that implements a deep machine learning model (hybrid convolutional neural network (CNN)-long-short-term-memory (LSTM) neural network) as an automatic diagnostic test presenting a test accuracy of 90 percent.
- The proposed networking strategy is implemented and tested for a total of 500 suspected patients. The delivered results show a polynomial computation complexity for the reporting time of the test results and a low storage overhead needed at both the healthcare entities and the suspected patient's device.

The next section describes the system architecture along with the desired functionality and security requirements.

Architecture and Objectives

System Model

The system consists of a certificate authority (CA), healthcare entities $\mathcal{C} = \{1, ...,C\}$ (e.g., hospitals and medical centers), users $\mathcal{N} = \{1, ...,N\}$ registered to the platform, and a dedicated node hosting the smart contract that implements the diagnostic test as shown in Fig. 1. The details are as follows:

- The CA generates and distributes the required keys. Moreover, it is the only entity that can identify any detected positive case for timely treatment. It does not have any copy of the ledgers and does not participate in any further activities.
- A healthcare entity $c \in \mathcal{C}$ has a copy of a ledger containing the collective symptoms of all suspected patients along with their diagnoses. The healthcare entities represent the main nodes in the consortium blockchain network, and hence, they must agree on the detection model hosted on the smart contract node. A consensus between all the entities in $\mathcal{C}$ must be reached in order to write or change anything in the smart contract.
- Whenever a registered user becomes a suspected patient, it sends the data of the symptoms to the smart contract node address for testing. Any registered user has a copy of a ledger containing only the test results for all anonymous suspected patients $\tilde{\mathcal{N}} \subset \mathcal{N}$.
- The Smart contract node receives the symptoms from each suspected patient and runs the detection model hosted on it for testing. The node then transmits two types of blocks over the network. The first block type is transmitted to the registered users, which includes the test results. The second block type is transmitted to the healthcare entities, which includes

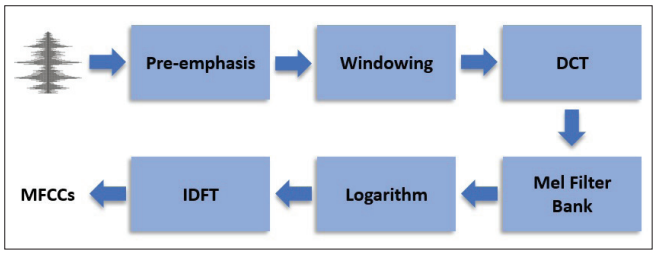

FIGURE 2. MFCCs feature extraction stages. DCT is discrete cosine transform and IDFT is inverse discrete Fourier transform.

the symptoms and the corresponding test results. Hosting a smart contract on a dedicated node (e.g., similar to [7]) adds flexibility to the system. This enables the developer to change the diagnostic test rules (detection model) if needed.

Threat Model

The privacy of the suspected patients is our main concern. Various entities (e.g., healthcare entities, other registered users, and eavesdroppers) follow the honest-but-curious adversary model. Hence, they aim to collect private information about the suspected patient. Such information includes the number of times the suspected patient took the test, the submitted symptoms, and the test results.

Platform Functionality and Security Objectives

The platform aims to support the following functionalities:

(F1) The suspected patient should be able to submit their symptoms at different time instants while preserving their privacy.
(F2) The suspected patient should be able to receive its diagnostic test result while preserving their privacy.
(F3) Only the identity of a positive case should be reported to a medical entity for timely treatment.
(F4) The healthcare entities should be able to have a database of symptoms and relevant diagnoses.
(F5) The healthcare entities should be able to update the detection model after achieving consensus.

The following security and privacy objectives should be satisfied:

(S1) User Anonymity: No entity should be able to link any test request to a specific registered user.
(S2) Data Unlinkability: No entity should be able to link any two testing queries to the same user.
(S3) Tamper-proof Messages: Previously published messages should not be changed by any entity.
(S4) Data Privacy: Each suspected patient should not have any access to the symptoms sent by other suspected patients.

The next section describes the details of the diagnostic test based on the deep machine learning model and its implementation in a smart contract. Then, the details of the proposed blockchain-based networking strategy are presented.

Deep Learning-Based Automatic Diagnostic Test

The developed model is implemented on the smart contract node for autonomous diagnostic. The objective is to develop a model based on simple signals collected by the suspected patients at home and submitted online for diagnosis. Recent research works have demonstrated that the coughing sound plays a vital role in identifying whether a suspected patient is infected or not with COVID-19 [8]. These works adopted different machine learning models such as support vector machine (SVM), logistic regression, decision trees, random forests, XG-BOOST, and deep transfer learning. The reported detection accuracy and F-1 scores range between 80 and 91 percent. The objective herein is to implement an accurate detection model in a smart contract node within the blockchain to assess the scalability of the online platform in terms of computational complexity and storage overhead.

DataSet

To develop the machine learning model, we used publicly available datasets Virufy COVID-19 [9] and COUGHVID [10]. The Virufy dataset contains 121 segmented cough samples with 16 positive COVID-19 cases. The COUGHVID dataset contains 20,000 records of cough samples; 1010 records of them are of COVID-19 positive cases. The COUGHVID dataset presents cases with a history of other respiratory conditions [10]. The clinical datasets are already labeled with positive and negative COVID-19 status.

Feature Extraction

In this article, two different approaches are investigated to extract indicative features that will be used to develop the detection model. The details are given below.

Approach-1: The audio file of the coughing sound is converted into a simple frequency spectrogram, which is then fed to the deep machine learning model.

Approach-2: The audio file of the coughing sound is converted into the Mel Frequency Cepstral Coefficients (MFCCs), which are then fed into the deep machine learning model. The reason for adopting such an approach is explained further. The coughing sound presents more energy in lower frequencies. The conversion of audio signals into MFCCs provides higher resolution in lower frequency using unequal spacing in the frequency bands. To extract the MFCCs for every sample, the steps shown in Fig. 2 are implemented.

Detection Models

Three deep machine learning models are investigated in this article to develop an accurate detection model. For each detection model, data has been split into 70 : 30 for training and testing, respectively. In addition, five-fold cross-validation has been adopted for hyper-parameter optimization. The details of the models are given below.

Deep CNN Model: A deep CNN model has been adopted due to its ability to efficiently extract local features from the input data. The developed optimal model consists of four hidden layers. Sixteen 3×3 kernels along with ReLU activation functions are used to obtain the feature map. The convolutional layers are followed by 2×2 max-pooling layers with stride as 1 and zero paddings. A dropout rate of 0.15 and Nadam optimizer are utilized.

Deep LSTM Model: This model has been adopted due to its ability to leverage temporal correlation within the data. Such ability enhances detection performance. The developed optimal model consists of three hidden LSTM layers that consist of 128 memory units. Again, the output layer implements a soft-max function, and a dropout rate of 0.15 and Nadam optimizer are utilized.

Deep CNN-LSTM Model: This hybrid model has been adopted so that the CNN part can extract local feature maps from the input data while the LSTM part can cover a longer temporal context of the feature maps. The developed optimal model consists of three convolution layers (with 32, 64, and 64 kernels of 3×3 size) followed by 2×2 max-pooling with stride as 1 and zero padding. The dropout rate adopted in this part is 0.1. Then, an LSTM layer with 128 memory units and a dropout rate of 0.3 is adopted. The output layer implements a soft-max function and a Nadam optimizer is utilized.

Detection Results

The detection results are compared in Table I. The best results are obtained when Approach-2 (based on MFCCs feature extraction) is adopted on a hybrid CNN-LSTM model presenting a detection accuracy of 90 percent. The world's experience with COVID-19 demonstrated an obvious trade-off between convenience, speed, and diagnostic test accuracy. For instance, the Polymerase Chain Reaction (PCR) test adopted in COVID-19 presents slow reporting of results (within 24 hours), and offers high accuracy of 95 to 100 percent. On the other hand,

Metrics	Precision			Recall			F1 Score			Accuracy		
Models	CNN	LSTM	CNN_LSTM	CNN	LSTM	CNN_LSTM	CNN	LSTM	CNN_LSTM	CNN	LSTM	CNN_LSTM
Approach-1	0.70	0.75	0.76	0.71	0.75	0.76	0.70	0.75	0.76	0.71	0.74	0.79
Approach-2	0.84	0.86	**0.90**	0.80	0.90	**0.91**	0.82	0.88	**0.90**	0.82	0.88	**0.90**

TABLE 1. Average detection results.

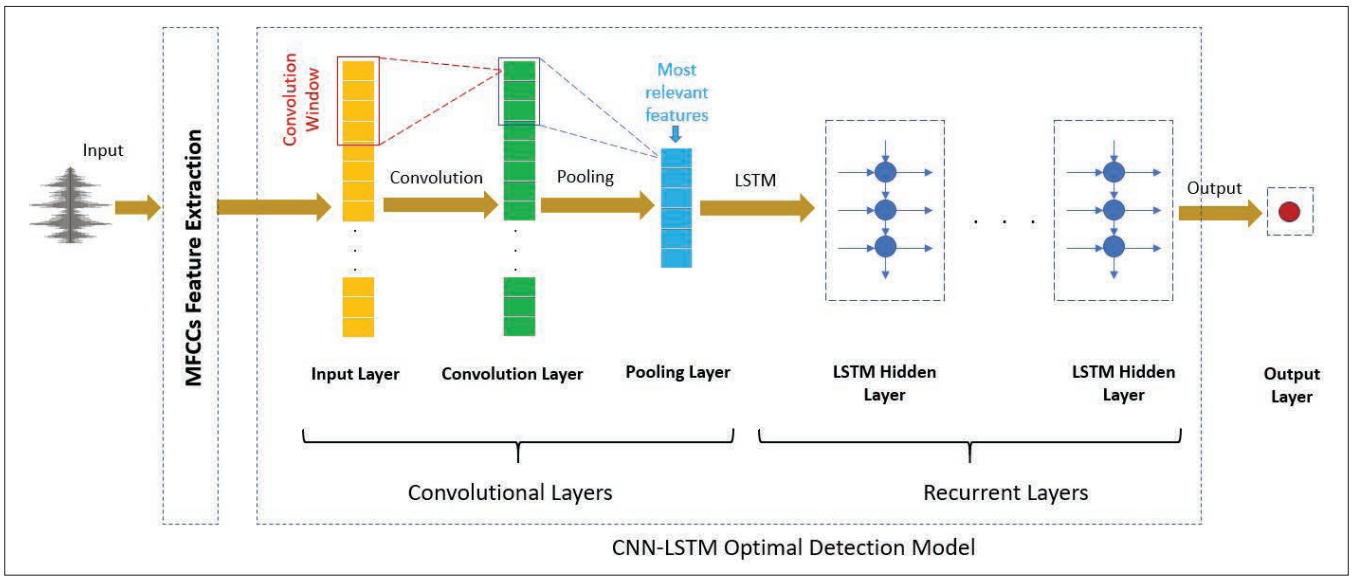

FIGURE 3. Abstraction of the operations implemented by the smart contract. The MFCCs feature extraction stages are detailed in Fig. 2.

the Antigen test presents faster reporting of results (within 30 minutes), offers an accuracy of 70 to 95 percent. Both tests require the suspected patient to go to the test center. The proposed platform is more convenient (can be taken at home), reports an accuracy of 90 percent (in the upper range of reported accuracy for diagnostic tests based on coughing sounds [8]), and we will show via experimental results that it can report the test results within 14 minutes for 500 suspected patients.

Smart Contract: Diagnostic Test Functionality

In light of the detection results, the diagnostic test functionality of the smart contract operates as follows. The process starts by extracting the MFCCs features from the cough samples, following the steps in Fig. 2. The extracted features are then fed to the detection model, which is based on CNN-LSTM architecture. Hence, the first part of the detection model adopts a set of convolution and pooling layers to extract some spatial features, as shown in Fig. 3. This is followed by a set of LSTM layers that extract temporal features. Finally, an output layer makes a decision on the test result. The data sharing and model update functionalities of the smart contract are explained in the next section.

Consortium Blockchain-Based Platform

Overview

The proposed blockchain network is based on a consortium implementation. The smart contract node is responsible for mining and broadcasting the blockchain. Hence, unlike public blockchains, a heavy consensus is not needed in the proposed platform. A light consensus model can be used instead. Overall, consortium and private blockchains sacrifice decentralization for efficiency, scalability, and safety. The proposed online platform operates based on five main phases, namely, initialization phase, test request phase, test report phase, data sharing phase, and model update phase, as shown in Fig. 1.

In the initialization phase, the CA generates and distributes all the necessary keys. In the test request phase, a suspected patient submits a transaction to the address of the smart contract node. The transaction includes the suspected patient coughing sound and a newly generated random number. The smart contract node runs the diagnostic test to determine the result. In the test report phase, a transaction is issued including the random number submitted previously by the suspected patient and the test result. A block is then generated by the smart contract node and broadcasted to the registered users to be saved in their ledger. In the data sharing phase, the symptoms data and the corresponding test result along with the random number are included in a transaction. A block is then generated by the smart contract node and broadcasted to the healthcare system entities to be saved in their ledger. The model update phase takes place only when the healthcare entities agree on an update to the parameters of the detection model. These phases are detailed next after presenting some preliminaries.

Distributed Random Number Generation

Relying on the public-private keys assigned to each registered user as data pointers does not provide data unlinkability. This is because all submitted data by the same registered user can be linked to the same key. In addition, the adoption of pseudo-random number generators (e.g., the sequences generated by natural number π, the linear congruence generator (LCG) pseudo-random numbers, etc.) does not solve the linkability issue as such numbers can be linked using simple neural network models [11]. Hence, we propose a unique random number that is generated on the fly on the device of each registered user each time a test request is submitted.

The proposed distributed random number generator ensures data unlinkability and the desirable collision-free feature of the generated random numbers among all users over time. The proposed random number is generated by user n at time t according to $R_{nt} = \text{HMAC}(K_{nt}, \Upsilon_n \,||\, Q_{nt})$, where HMAC is the keyed-hash message authentication code (HMAC) with SHA-1 hash, K_{nt} is a pseudo-random number, Υ_n is the private key of user n, and Q_{nt} is a counter. Using only K_{nt} as the random number identifier creates two issues: (a) collision probability and (b) the possibility of linkability using neural networks. So, instead of using K_{nt} as the identifier, it is used as the key to the HMAC function. The user's private key Υ_n is used as a part of the HMAC message to ensure there is no collision probability. The private key Υ_n is concatenated with a counter value Q_{nt} to break any periodicity in the generated random number. This process ensures the generation of a unique decentralized untraceable random number.

Details of the Operation Phases

In the following, we explain the details of the five operation phases of the proposed online platform.

Initialization Phase: It takes place while setting up the network. First, the users register to the platform. Based on the number of registered users, the CA uses the group signature scheme introduced in [12] to generate a set of keys. These keys are (a) group key κ that is distributed to the healthcare entities and registered users; (b) a set of N private keys where each registered user receives a private key (Υ_n for user $n \in \mathcal{N} = \{1, ..., N\}$); and (c) a private key χ that is kept only with the CA, which can be used to reveal the real identity of the signer of any given message. This is needed only when a positive case is detected where its identity is forwarded to a hospital or a medical center so that the patient receives timely treatment. In addition, the CA generates a set of public/private key pairs (PK_c, SK_c) for healthcare entity $c \in \mathcal{C} = \{1, ..., C\}$) and a public/private key pair for the smart contract node (PK,PS). The CA does not have any access to any ledger in the blockchain network.

Test Request Phase: In this phase, a suspected patient $n \in \check{\mathcal{N}}$ sends a test request to the address of the smart contract node. The test request transaction, Ω_{nt}, includes the symptoms data (audio signal of the coughing sound D_{nt} for user n at time t) that will be applied to the detection model for the suspected patient whose generated random number at time t is R_{nt}. This transaction is signed using the group key κ and the private key of the suspected patient Υ_n, resulting in the signature σ_{nt}. Hence, $\Omega_{nt} = R_{nt} \,||\, D_{nt} \,||\, \sigma_{nt}$. Upon receiving the test request, the smart contract node checks the signature to ensure that the sender is a legitimate user of the platform. Then, the smart contract node uses the detection model to specify the diagnostic test result Y_{nt}, where $Y_{nt} = 0$ indicates a negative case while $Y_{nt} = 1$ indicates a positive case. Hence, the platform has successfully performed functionality (F1).

Test Report Phase: In this phase, the smart contract node generates a transaction, Ψ_{nt}, for each suspected patient. Each transaction includes the random number sent in the request R_{nt} and the corresponding test result Y_{nt}. The transaction is signed by the private key of the smart contract node. Hence, $\Psi_{nt} = R_{nt} \,||\, Y_{nt} \,||\, \sigma_{\text{SK}}$. The smart contract node waits until a specific number of transactions is generated and creates a test report block that is broadcasted to all of the registered users. The test report block $\Psi_{\tilde{I}}$ includes:

- Index ($\tilde{I}$): The block number in the chain.
- Previous Hash ($\text{H}_{\tilde{I}-1}$): The hash of the previous block.
- Transactions (Ψ_{nt}): Including the test results for the suspected patients $n \in \check{\mathcal{N}}$ that requested a test at t.
- Timestamp ($\tilde{t}$): The time when the block is generated.

For a suspected patient with $Y_{nt} = 1$, a request is submitted to the CA to identify the patient and contact the nearest healthcare entity to provide the necessary treatment in a timely manner. Hence, the platform has performed functionalities (F2) and (F3).

Data Sharing Phase: The smart contract node prepares another block of transactions to share the symptoms data and their diagnoses with the healthcare entities. Each transaction includes a random number R_{nt}, the corresponding symptoms data D_{nt}, and the diagnostic result Y_{nt}. This transaction is signed by the private key of the smart contract. Hence, the data sharing transaction is given by $\Gamma_t = R_{nt} \,||\, D_{nt} \,||\, Y_{nt} \,||\, \sigma_{\text{SK}}$. A specific number of transactions is then collected together to form a block, which will be broadcasted to all the healthcare entities. Hence, the data sharing block is given by $\Gamma_{\tilde{I}} = \tilde{I} \,||\, H_{\tilde{I}-1} \,||\, \Gamma_n \; \forall n \in \{1, ..., \check{\mathcal{N}}\} \,||\, \hat{t}$. Hence, the platform has performed functionality (F4).

Model Update Phase: The data in the block broadcasted to the healthcare entities can be used to update the detection model. Any healthcare entity c can suggest a modification to the parameters by encrypting the new model parameters P_t^* using its private key SK_c and sends it to the rest of the participating healthcare entities and the smart contract node. The health entities decrypt the received message and extract the new parameters P_t^* and check them. If the healthcare entity agrees on the new parameters P_t^*, it encrypts it using its private key SK_c and sends it to the smart contract node. The smart contract node decrypts the messages using the public key of the healthcare entity PK_c and extracts the new parameters P_t^*. If all the healthcare entities send the same new parameters P_t^*, the smart contract will update the model with the new parameters. Hence, the platform has performed functionality (F5).

Experimental Results

Implementation Details: Existing platforms (e.g., the Hyperledger Fabric) cannot be used to evaluate the performance of the proposed platform as they are not flexible enough to allow us to make modifications in the blockchain design to introduce the proposed random number identifiers and group signatures. Hence, the proposed platform is implemented from scratch using an Ubuntu operating system on a virtual machine (similar to [13]) with a 1.8 GHz processor and a RAM of 7 GB. All the cryptographic materials are tested using the charm crypto library. A case study of three healthcare entities (hospitals) is implemented with up to 500 suspected patients requesting a diagnostic test at time t. In the implementation, different computer ports are used to mimic different entities as follows: (a) ports (5000 to 5002) are dedicated to the hospitals; (b) port 5003 is dedicated to the smart contract node; and (c) ports (5004 to 5504) are dedicated to the users. In the user's node file, an extension "/transaction" is implemented using a Python code to prepare and send the transaction including the random number identifier, coughing sound signal, and signature to the address "5003/MLmodel," which is the address of the smart contract node running the diagnostic test model. The Python code in the smart contract's node file includes an extension called "/MLmodel." In this extension, the signature is first verified. If it passes the verification, the coughing signal is extracted from the transaction and fed to the diagnostic test model. A number of transactions are then grouped to form a block. This block contains the block number, the hash of the previous block, and the set of transactions. The generated blocks are then broadcasted to the users and the hospitals.

Performance Evaluation: The performance is evaluated by measuring the block generation time and the ledger size. These metrics are commonly used in literature to measure the performance of a blockchain network [14, 15].

The block generation time is the time taken by the smart contract node to run the detection model and broadcast both the test results and symptoms data with the corresponding diagnosis in separate blocks to the registered users and hospitals, respectively. The block generation time is shown in Fig. 4, which demonstrates a linear time complexity versus number of submitted test requests. For instance, it takes roughly 14 minutes to provide test reports for 500 suspected patients who submitted simultaneous test requests. This presents a good balance between con-

venience (for taking the test at home/online) and waiting time (as it is less than the required waiting time for the Antigen rapid test).

The data storage overhead is the storage needed at each registered user and hospital to save the blockchain ledger. Fig. 5 presents the storage overhead versus the number of submitted test requests. The storage size grows with the number of submitted test requests. The storage at the hospital's ledger is larger than that of the user's ledger as it accounts for the size needed to store the symptoms. Nevertheless, as shown in Fig. 5, low storage is required, which is up to 0.52 MB and 0.6 MB at the user and hospital side, respectively.

Security and Privacy Analysis: The proposed strategy gains its security from the group signature, the random numbers, and the blockchain technology. The group signature ensures both the anonymity of the users and the unlinkability of the data, hence satisfying (S1) and (S2). The random number generator points a specific user to its data without revealing its identity or symptoms to other registered users or linking its data over time. The blockchain helps in making the data unchangeable over time and distributed to all users and healthcare entities, hence satisfying (S3). The presence of one ledger for the healthcare entities and another ledger for the suspected patients helps in providing access control. The ledger distributed to all the healthcare entities (which includes the symptoms and the corresponding test results) helps in further improving the detection model. The suspected patient's ledger has only the random numbers and the corresponding test results and does not include any symptoms data to protect the privacy of the suspected patients, satisfying (S4).

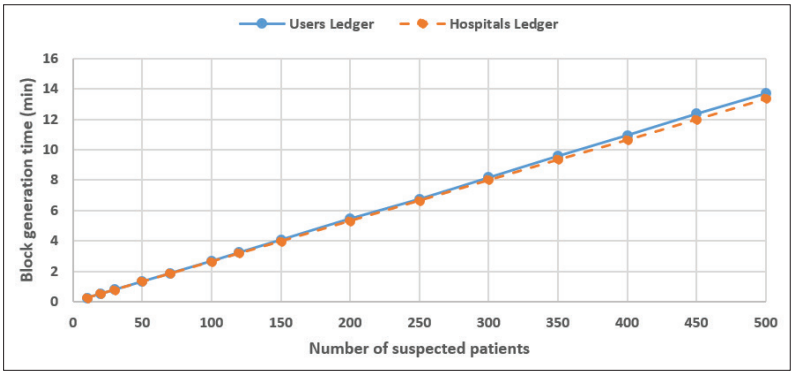

FIGURE 4. Illustration of the block generation time for the two separate ledgers.

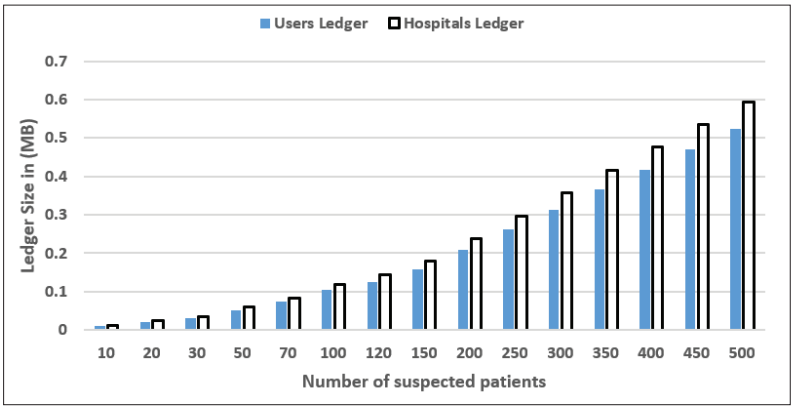

FIGURE 5. Illustration of storage overhead for the users and hospital ledgers.

Conclusions

This article presented an online automated platform that can remotely collect the symptoms needed from suspected patients and provide a fast diagnosis. The proposed platform preserves the privacy of the users in terms of both anonymity and data unlinkability while offering 90 percent detection accuracy based on coughing signals. The platform exhibits low complexity as it takes roughly 14 min for any user in a group of 500 users to know its test result. In addition, the storage requirements for each user and healthcare entity are 0.52 MB and 0.6 MB, respectively. Future extensions of the proposed platform will consider the presence of malicious nodes such as users, hospitals, and/or smart contract nodes.

References

[1] W. B. Karesh *et al.*, "Wildlife Trade and Global Disease Emergence," *Emerging Infectious Diseases*, vol. 11, no. 7, 2005, p. 1000.

[2] H. R. Hasan *et al.*, "Blockchain-based Solution for COVID-19 Digital Medical Passports and Immunity Certificates," *IEEE Access*, 2020.

[3] H. Xu *et al.*, "Beeptrace: Blockchain-enabled Privacy-preserving Contact Tracing for COVID-19 Pandemic and Beyond," *IEEE Internet of Things Journal*, vol. 8, no. 5, 2021, pp. 3915–29.

[4] K. Yu *et al.*, "Efficient and Privacy-preserving Medical Research Support Platform against COVID-19: A Blockchain-based Approach," *IEEE Consumer Electronics Mag.*, vol. 10, no. 2, March 2021, pp. 111–20.

[5] R. Kumar *et al.*, "Blockchain-federated-learning and Deep Learning Models for COVID-19 Detection Using CT Imaging," *IEEE Sensors Journal*, vol. 21, no. 14, 2021, pp. 16301–14.

[6] M. A. Rahman *et al.*, "Secure and Provenance Enhanced Internet of Health Things Framework: A Blockchain Managed Federated Learning Approach," *IEEE Access*, vol. 8, 2020, 2020, pp. 205071–87.

[7] Blockone Developers, "Eosio." Available on: https://github.com/ EOSIO. Accessed: 2021-04-09.

[8] C. Brown *et al.*, "Exploring Automatic Diagnosis of COVID-19 from Crowdsourced Respiratory Sound Data," in *Proc. 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3474–84.

[9] Virufy Dataset. Available on: https://virufy.org/index.html. Accessed: 2021-04-09.

[10] COUGHVID Dataset. Available on: https://coughvid.epfl.ch/. Accessed: 2021-04-09.

[11] Y. Feng and L. Hao, "Testing Randomness Using Artificial Neural Network," *IEEE Access*, vol. 8, 2020, pp. 163685–93.

[12] D. Boneh *et al.*, "Short Group Signatures," in *Proc. Annual International Cryptology Conference*, Springer, 2004, pp. 41–55.

[13] DAPP University Team, "Blockchain Python Programming Tutorial." Available on: https://www.youtube.com/watch?v=pZSegEXtgAE, Oct 2019. Accessed: 2021-04-09.

[14] Hyperledger Fabric Team, "Hyperledger blockchain performance met.rics white paper." Available on: https://www.hyperledger.org/learn/ publications/ blockchain-performance-metrics, Oct 2018. Accessed: 2021-04-09.

[15] Z. Dong *et al.*, "DAGbench: A Performance Evaluation Framework for DAG Distributed Ledgers," in *Proc. 2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, 2019, pp. 264–71.

Biographies

Mahmoud Abouyoussef received the B.Sc. degree (Hons.) in electronics and communication engineering from Misr International University, Cairo, Egypt in 2012. He received two M.Sc. degrees from Mid Sweden University and Misr International University in 2012 and 2019, respectively. He is currently pursuing a Ph.D. degree with the Computer Science Department, Tennessee Tech Univ., USA.

Surbhi Bhatia received the Ph.D. degree in computer science and engineering from Banasthali Vidypaith, India. She is currently an assistant professor in the Department of Information Systems, College of Computer Sciences and Information Technology, King Faisal University, Saudi Arabia.

Poonam Chaudhary received the B.Tech. degree from the University of Rajasthan, Jaipur, and the M.Tech. degree from Mahrishi Dayanand University Rohtak, Haryana. Currently, she is pursuing a Ph.D. degree from MRIIRS Faridabad.

Saurabh Sharma is pursuing a B.Tech. degree from NorthCap University, Gurugram, India.

Muhammad Ismail received the B.Sc. (Hons.) and M.Sc. degrees in electrical engineering from Ain Shams University, Egypt, in 2007 and 2009, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Canada, in 2013. He is an assistant professor with the Computer Science Department, Tennessee Tech. University, USA. He received the best paper awards from IEEE IS'20, IEEE TCGCN in IEEE ICC'19, IEEE Globecom'14, IEEE ICC'14, Green'16, SGRE'15. He is an associate editor with IEEE IoTJ and IEEE TGCN.