



Computer-Aided-Diagnosis as a Service on Decentralized Medical Cloud for Efficient and Rapid Emergency Response Intelligence

Amirhossein Peyvandi¹ · Babak Majidi^{1,2}  · Soodeh Peyvandi³ · Jagdish Patra⁴

Received: 19 December 2020 / Accepted: 19 June 2021 / Published online: 27 June 2021
© Ohmsha, Ltd. and Springer Japan KK, part of Springer Nature 2021

Abstract

The COVID-19 pandemic resulted in a significant increase in the workload for the emergency systems and healthcare providers all around the world. The emergency systems are dealing with large number of patients in various stages of deteriorating conditions which require significant medical expertise for accurate and rapid diagnosis and treatment. This issue will become more prominent in places with lack of medical experts and state-of-the-art clinical equipment, especially in developing countries. The machine intelligence aided medical diagnosis systems can provide rapid, dependable, autonomous, and low-cost solutions for medical diagnosis in emergency conditions. In this paper, a privacy-preserving computer-aided diagnosis (CAD) framework, called Decentralized deep Emergency response Intelligence (D-EI), which provides secure machine learning based medical diagnosis on the cloud is proposed. The proposed framework provides a blockchain based decentralized machine learning solution to aid the health providers with medical diagnosis in emergency conditions. The D-EI uses blockchain smart contracts to train the CAD machine learning models using all the data on the medical cloud while preserving the privacy of patients' records. Using the proposed framework, the data of each patient helps to increase the overall accuracy of the CAD model by balancing the diagnosis datasets with minority classes and special cases. As a case study, the D-EI is demonstrated as a solution for COVID-19 diagnosis. The D-EI framework can help in pandemic management by providing rapid and accurate diagnosis in overwhelming medical workload conditions.

Keywords Emergency response · COVID-19 pandemic · Blockchain · Computer aided diagnosis · Machine learning

✉ Babak Majidi
b.majidi@khatam.ac.ir

Extended author information available on the last page of the article

Introduction

During epidemiological events, such as the COVID-19 pandemic, the emergency services face a significant number of rapidly deteriorating patients with severe conditions and high chances of mortality. The computer-aided diagnosis (CAD) systems can help the emergency services to have an initial estimate of the patient's conditions and to provide rapid response to a large number of patients. Artificial intelligence and machine learning are utilized for many applications in management of various aspects of smart environments and privacy preserving [1–3]. These algorithms and models can also be used for automated emergency response to pandemic conditions. Some of the proposed applications include early risk assessment [4], risk prediction in patients [5], clinical trials for drugs [6], measuring the effects of contamination [7] and allocation of priorities in receiving the diagnosis tests in areas with shortage of supply [8]. Furthermore, deep learning has been used for training different models for autonomous identification of infected patients by analyzing computed tomography (CT) and X-ray images [9] and to detect infected patients by analyzing their cough pattern [10]. The CAD based telemedicine can help patients with limited symptoms to receive the required diagnosis and treatment at home and only to visit hospitals in cases when their condition deteriorates.

For designing accurate machine learning based CAD systems, the researchers require a significant amount of data for modeling the symptoms of the disease. However, training accurate medical machine learning models requires high-quality data which contains enough samples from each category and specific cases. These data improves the machine learning models and allow them to identify minority classes with higher accuracy. However, a larger part of the data that can help the researchers to model the diseases is not publicly available as privacy does not allow sharing the patient data. Therefore, the subject of privacy-preserving for medical applications is of significant interest for researchers. The blockchain technology is one of the most prominent technologies proposed for privacy-preserving and protecting medical records. Recently, blockchain technology is integrated into different fields, scenarios, and environments. Examples of this integration include data sharing for the Industrial Internet of things (IIoT) [11], large-scale heterogeneous network (LS-HetNet) and 5G-enabled smart cities [12]. This technology has been used for securing healthcare data in various recent studies. For example, Gupta et al. [13], Healthify [14], Bittins [15] proposed blockchain-based solutions for secure medical and healthcare data transfer and sustaining trust while satisfying data exchange requirements within the healthcare ecosystem. Furthermore, privacy protection in IoT services has been investigated [16]. In addition to secure data transfer, the blockchain is also proposed for secure training of machine learning models. Shen et al. [17] proposed secureSVM, a privacy-preserving Support Vector Machine (SVM) training scheme over blockchain-based encrypted internet of things (IoT) data. In their proposed framework, two main entities exist: (1) IoT Data Provider and (2) data analyst. Each data provider pre-processes IoT data instances, encrypts them locally using their private

keys, and records them in a blockchain-based shared ledger. The data analyst who wants to train an SVM model can get access to the encrypted data. In this framework, during the training process, interactions between the data analyst and data providers are crucial for exchanging intermediate results. PrivySharing [18], is a privacy-preserving framework focused on secure IoT data sharing in a smart city environment. The proposed strategy ensures that personal/critical user data is kept confidential, securely processed, and is exposed to the stakeholders on a need-to-know basis as per user-defined Access Control List (ACL) rules embedded in smart contracts. Chen et al. [19] proposed a Blockchain-based secure inter-hospital system, which was used to share electronic medical records (EMRs) between different hospitals. Their proposed scheme is immune to data manipulation and data leak. Dai et al. [20] proposed a blockchain-enabled Internet of Medical Things (IoMT) to address the security and privacy concerns of IoMT systems. Shu et al. [21] proposed a certificate-less scheme, which offers secure storage for sharing medical data on the blockchain. They stated that their scheme would satisfy the security requirements (integrity, privacy, and traceability) in medical cyber-physical systems (MCPS). In their work, the scheme consists of two layers and the medical records are stored off-chain. Jaleel et al. [22] proposed a medical data sharing framework by collaborating with different IoMT devices. Since devices created by different vendors offer different data types and formats, this framework aims to fix incompatible data, merge them, and finally publish them. Moreover, the blockchain has also been proposed to be utilized for research regarding COVID-19 pandemic [23]. Fadaeddini et al. proposed a framework for training machine learning models on decentralized blockchain network [24, 25].

To utilize the security of decentralized blockchain-based networks for medical emergency systems, in this paper, we propose a CAD framework that allows the collaboration of different clinics, hospitals, and medical centers during pandemics. The proposed framework, called Decentralized deep Emergency response Intelligence (D-EI), uses a combination of blockchain and machine learning for emergency CAD. The proposed decentralized framework guarantees data integrity and patient privacy. As a case study, the applications of the D-EI framework is investigated for several COVID-19 diagnosis criteria and a machine learning based CAD using D-EI is proposed. The electro-cardiograms (ECG), computed tomography (CT) scans, X-ray images and cough recordings of the patients are used in this investigation. The experimental results show that the proposed framework can provide a highly effective solution for rapid and privacy preserving CAD in pandemic conditions.

The rest of the paper is organized as follows: In “Decentralized Deep Emergency Intelligence”, D-EI framework is detailed. “D-EI for Emergency Intelligence in COVID-19 Pandemic” is dedicated to D-EI as a solution for privacy preserving COVID-19 diagnosis. Finally, “Conclusion” concludes the paper.

Decentralized Deep Emergency Response Intelligence

The sharing of medical test results and other personal data is required for management of public health emergencies during pandemics. This requirement raises issues regarding the security of data and the privacy of the patients. Blockchain and smart contracts can provide a solution for these issues. In the decentralized approach, data remains distributed on individual devices, and therefore, data will remain anonymous. This approach is opposite to the centralized approach in which the data is aggregated on a central server, and at best, the data can be pseudonymized. Using the D-EI framework, patients can share their test results between different hospitals and medical centers without any concern regarding their privacy and accurate machine learning models can be trained using these data.

Decentralized Privacy-Preserving Framework

A smart contract is a transaction protocol which is executed automatically and performs different functions such as funds transfer, calculations, and information storage on a blockchain [26]. The smart contract includes the value, address, functions, and state of the transactions executed between the participating nodes. It accepts transactions as input, executes the code, and provides the output. Notable characteristics of smart contracts are accuracy, immutability, security, self-enforcement, fast execution, and self-verifiability. In other words, smart contract is a programming code which allows decentralized automation. Upon execution, the smart contract executes itself and enforces the agreement conditions. Usually, the smart contracts are developed in a programming language called Solidity over various blockchain networks such as Stellar, Ripple, Ethereum, and EOS. These platforms ensure that data will be protected against cyber-security attacks such as modification, spoofing, and fabrication attacks.

Although the smart contracts are considered secure, there are still vulnerabilities which should be considered in development of smart contract based applications [27]. An example of security vulnerabilities in Ethereum smart contracts is *reentrancy* attack which may happen when a smart contract calls an external contract that takes over the control flow and calls back into the calling contract before the first invocation is finished. This process can lead to improper initialization, mishandled exceptions, unprotected self-destruction, and delegation calls to untrusted contracts [27]. However, there are solutions for these vulnerabilities. For example, *reentrancy* can be solved by avoiding calling an external function until all of the internal functions are finished. Alternatively, developers can avoid security vulnerabilities by removing the selfdestruct function. Furthermore, there are many formal verification techniques (Formalization based on Lem or F^* [28]), code analysis tools (Securify [29], Zeus [30]), or security audits (OpenZeppelin [31], SmartDec [32]) that developers can perform before deploying the smart contract, to eliminate security weaknesses. Moreover, security patterns can be applied in the design phase from the beginning. Therefore, even though smart contracts are not immune to vulnerabilities, security issues can be handled and smart contracts can be verified using

different tools and techniques. For Solidity programming language, used in the proposed D-EI, a satisfiability modulo-theories (SMT)-based formal verification module is utilized to verify the smart contracts [33]. This verification tool is integrated into the Solidity compiler, and during compilation, issues warnings for potential failures.

In the proposed D-EI framework, there are two groups of stakeholders and one controller. The first stakeholder is a patient. The health records of the D-EI framework members can be distributed across emergency response teams, clinics and hospitals for usage by the patient and for training CAD models. In emergency situations the patients require to have access to all of their medical records, wherever they are. Since the focus of the current study is on the COVID-19 pandemic, we assume these records are COVID-19 test results. However, the D-EI is also applicable in other medical and healthcare scenarios. The second group of D-EI stakeholders are medical centers. The medical centers store the healthcare data, medical records, and test results of patients. The D-EI controller is a smart contract, which is responsible for storing and accessing records on a blockchain network and handling permissions and access. Moreover, the smart contract encrypts the values before creating a transaction and stores them on the blockchain. It will also be responsible for storing files, if necessary. Finally, the smart contract is utilized for machine learning based CAD and delivers the final test result.

Figure 1a shows the overall architecture of the proposed D-EI framework. The patient medical information (such as medical images and lab results) are stored in a decentralized manner on the blockchain using InterPlanetary File System (IPFS) [34]. IPFS is a decentralized peer-to-peer protocol which distributes files between a list of trusted nodes known as bootstrap nodes and makes them available to other users by Content Identifiers (CIs) and content-based addressing. It takes advantage of cryptographic hashes to store files on a blockchain network. Figure 1b, c show COVID-19 CAD using D-EI. First, a patient takes an RT-PCR test (Step 1), and the result is added to the system by a nurse via the smart contract (Step 2). Then, the smart contract will create a transaction, and this value will be stored on the blockchain (Step 3). In this scenario, since we will only work with binary results (either 0 or 1, for negative or positive results), there is no need for file storage.

Later, if other tests are taken, their result will be added to the system (steps 7 and 8) using a simple weighted average fusion method, and the final result will be shown to the patient (steps 9 and 10). The raw results will be a list in the format of “[Test name, Result]”. For example, [PCR, 1] means the result of RT-PCR is positive or [Cough, 0] means the cough test result is negative. The patient data can also be used for training the CAD models using similar method.

D-EI Smart Contract

In the proposed D-EI framework, a single Ethereum-based [35] smart contract is utilized. It is used to manage data sharing between medical centers and patients and to receive test results and perform diagnosis based on results. Before connecting to the smart contract, patients and clinics are required to set up a crypto

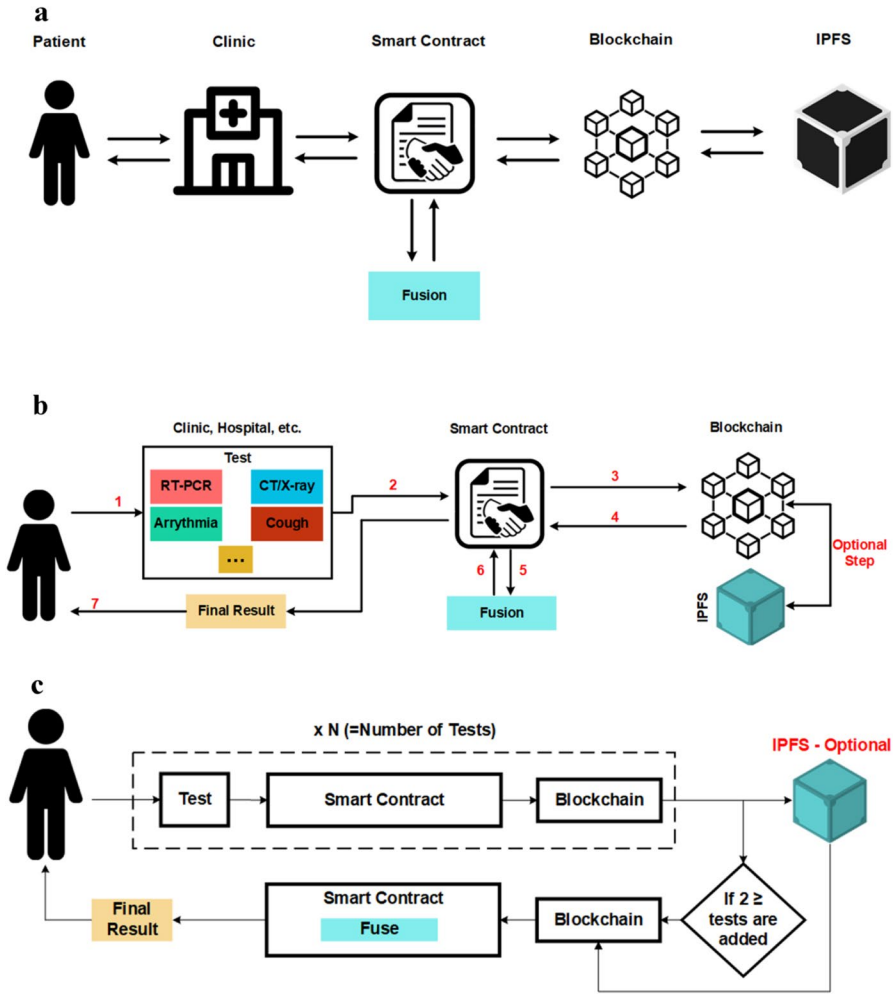


Fig. 1 Decentralized framework for sharing data between different data holders for telemedicine

wallet or a blockchain browser. By doing so, they will be given a unique blockchain address, which will act as an identifier and distinguishes users from each other. Moreover, in order to re-use this address and to connect to it using other apps, browsers, or wallets, they are given a private key or mnemonic phrase. MetaMask [36] is utilized for this task in the proposed D-EI framework. The address given to the patient (for example, *0xBe3F06...Fd67993c*) is unique and can only be used by the patient. For programming the Ethereum-based smart contract, Solidity 0.5.0 [37] is employed. Compiling the smart contract is done using Truffle Suite [38], and the migration for development on a local blockchain and further evaluation and tests, was achieved by Ganache [39]. The connection between

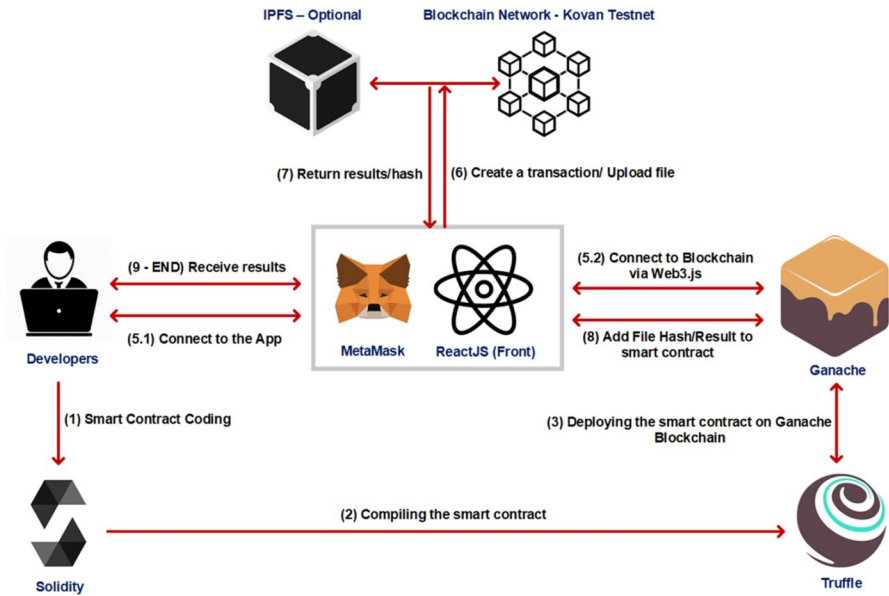


Fig. 2 Deployment and test phases of D-EI

the front-end and blockchain is controlled by web3.js [40], an Ethereum JavaScript API. Figure 2 shows the deployment and test phases.

Restricting access to unauthorized users is vital in smart contracts. In order to manage access control and security of the smart contract, OpenZeppelin, a security tool for Solidity, was utilized. For registering in the proposed D-EI system, first, clinics and hospitals must access the application and register with their unique address given by MetaMask or any other crypto wallet. Upon registration, they will be given *ClinicRole*, which is an administrator role for that specific clinic. After that, when this clinic adds new patients to the system, these patients will be assigned a *PatientRole*, and they will be linked to their corresponding clinic. Each patient can also be linked to different clinics since each clinic or hospital has a unique address. Therefore, a patient’s distributed medical records will be accessible to him in a decentralized and secure manner. This was achieved by employing OpenZeppelin’s *AccessControl* module [41], which assigns different roles to different system stakeholders while taking care of permissions and secure access. At first, the patients must create a wallet address. Then, for registration, they must contact the medical centers and provide their blockchain address for the record. After the clinic or hospital verifies them (via their insurance card, ID card or passports), the clinic or hospital can add the patient to the smart contract via their address, and the *Patient* role will be assigned to them. Moreover, only accounts that have been granted the *ClinicRole* can grant or revoke *PatientRole* for an account that is linked to them. Algorithm 1 and Algorithm 2 demonstrate how OpenZeppelin’s *AccessControl* module was employed in the smart contract, how the roles are granted to stakeholders of the system and how new users and admins can be registered and added. Here, two roles

(*clinicRole* and *patientRole*) are defined. In OpenZeppelin, roles are referred to by their “bytes32” identifier (Algorithm 1 – lines 1 and 2). The address passed on to the constructor is the initial administrator.

Algorithm 1: Adding necessary roles and access control functions to the system

Input: Stakeholder’s addresses

Output: Successful registration of a clinic with correct role

- 1 Set bytes32 public constant *clinicRole* = keccak256(“*ClinicRole*”)
 - 2 Set bytes32 public constant *patientRole* = keccak256(“*PatientRole*”)
 - 3 **constructor** *setupRoles*(*address clinicAddress*):
 - 4 | _*setupRole*(*clinicRole*, *clinicAddress*)
 - 5 | _*setRoleAdmin*(*patientRole*, *clinicRole*)
 - 6 **Function** *isClinicAdmin*(*address acc*):
 - 7 | **return** *hasRole*(*clinicRole*, *acc*)
 - 8 **Function** *isPatient*(*address acc*):
 - 9 | **return** *hasRole*(*patientRole*, *acc*)
-

The *_setupRole* and *_setRoleAdmin* (Algorithm 1 – lines 4 and 5) are OpenZeppelin functions. The *_setupRole* takes a role and address as its arguments and creates a new role linked to its creator’s address (*clinicAddress*). The *_setRoleAdmin* function links two roles and by default *clinicRole* is the admin role of *patientRole*. The *isClinicAdmin* and *isPatient* functions, which make use of OpenZeppelin’s pre-defined *hasRole* function, will be used to handle access control throughout the entire smart contract, and they return a Boolean value (True or False).

Algorithm 2 represents the registration of new admins (head of a department, a doctor, or a nurse) for the same clinic and its patients. Here, we make use of the previously defined *isClinicAdmin* function in Algorithm 1. The *grantRole* function is also a part of OpenZeppelin *AccessControl* module. This function utilizes the implicitly available *msg.sender* from global variable *msg*, which contains the address of the stakeholders, and the user who is currently connected to the system.

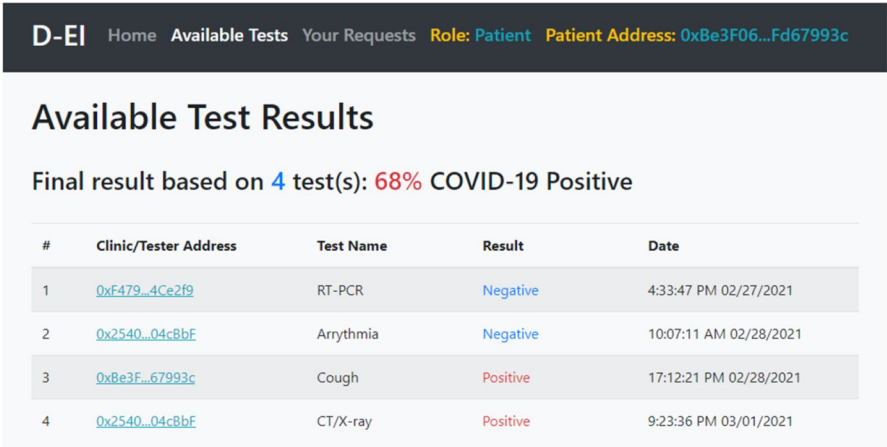


Fig. 3 A patient viewing his/her COVID-19 test results added to the system

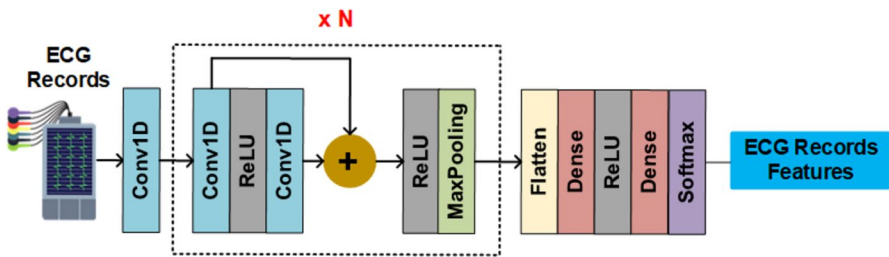


Fig. 4 The architecture of the deep neural network of Arrhythmia classification model

Algorithm 2: Registering new admins and new patients for a clinic

Input: A clinic or hospital’s address

Output: Successful registration of new admins and users with right permission and access

```

1 Function addClinicAdmin(address acc):
2   if isClinicAdmin(msg.sender) then
3     | grantRole(clinicRole, acc)
4   end
5 Function addPatient(address acc):
6   if isClinicAdmin(msg.sender) then
7     | grantRole(patientRole, acc)
8   end

```

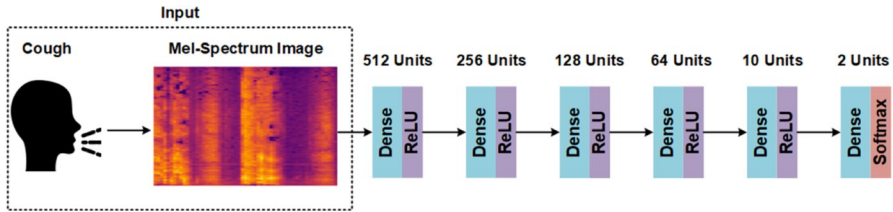


Fig. 5 The architecture of the deep neural network of COVID-19 cough classification model

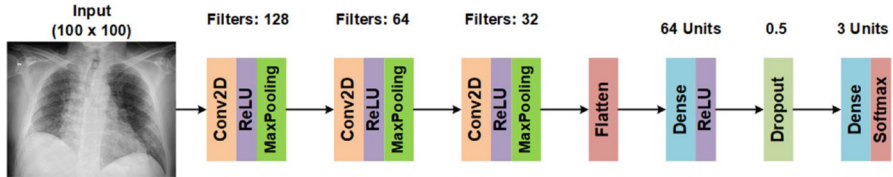


Fig. 6 The architecture of deep neural network of X-ray classification model

After the smart contract tested locally, it was deployed on an online blockchain named Kovan Testnet Network [42] for further evaluation. Furthermore, for uploading files on IPFS and later downloading them, an IPFS API named Infura [43] was employed. Figure 3 shows the UI of the D-EI system, accessed by a patient who is viewing his/her COVID-19 tests result. As shown, the Cough and CT/X-ray tests were positive, and after applying the CAD, the final result is shown. This data can also be used for training CAD machine learning models.

Machine Learning-Aided COVID-19 CAD Framework

For demonstration of the machine learning-based COVID-19 CAD on D-EI, the application of deep learning methods for Arrhythmia classification, cough detection, chest CT segmentation, and X-ray classification are investigated in this paper. For the Arrhythmia classification, a convolutional neural network, as shown in Fig. 4, is utilized. In this architecture, all convolution layers apply 1D convolution with 32 kernels of size 5. Max-pooling of size 5 and stride 2 are used in all pooling layers. The predictor network consists of five residual blocks followed by two fully-connected layers with 32 neurons each. Finally, a *Softmax* layer is used to predict output class probabilities. Each residual block contains two convolutional layers, two rectified linear unit (ReLU) activation function, a residual skip connection, and a pooling layer.

The cough classification model is illustrated in Fig. 5. This architecture employs densely connected layers, and each dense layer uses a ReLU activation function. Same as before, the *Softmax* activation function is used in the last layer. First, the input of this architecture is a recorded cough, save as an audio file. However, before feeding it to the input, the sound must be converted into a Mel-Spectrum format, and then it can be sent to the first dense layer.

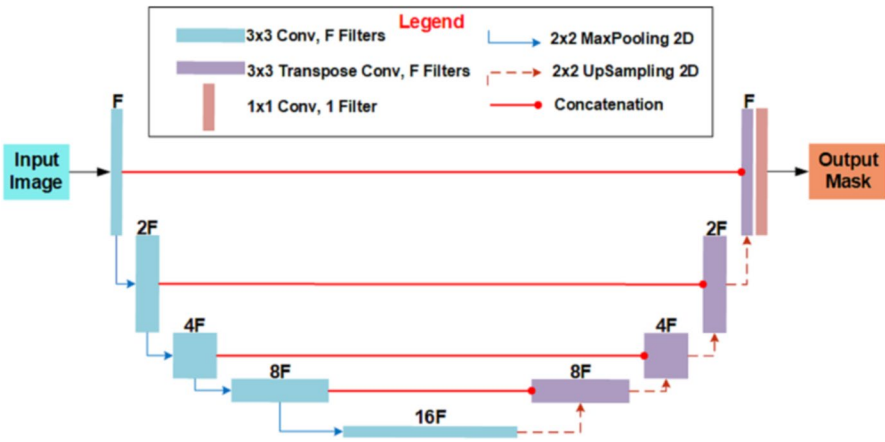


Fig. 7 The architecture of U-Net for chest CT segmentation

The X-ray classification model makes use of convolutional layers, as shown in Fig. 6. In this model, the input X-ray images are resized to 100×100 pixels. A series of *Conv2D* layers with a kernel of size 3, ReLU activation function, and *MaxPooling2D* layers with a pool size of 2 are employed. The convolutional layers output is then passed to a Flatten layer, with passes its output to a dense layer with 64 units. Next, a *Dropout* with a value of 0.5 is applied in order to prevent overfitting. Finally, a dense layer with a *Softmax* activation function is used as the final output.

For CT image segmentation, the U-Net architecture, illustrated in Fig. 7, is employed. The U-shaped network is comprised of two different paths: a contracting path and an expansive path. The contracting path is a convolutional network that consists of repeated application of 3×3 convolutions, each followed by a ReLU and a filter of size F , which in this case study was set to 32. Furthermore, *MaxPooling2D* and *Dropout* were employed in the contracting path. The expansive path merges the feature and spatial information through transposing and concatenations of the features from the contracting path.

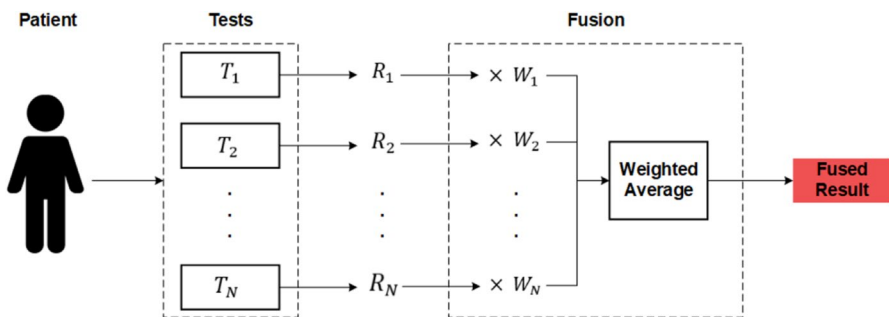


Fig. 8 The overall fusion method used in this paper

A problem with combing the results of multiple deep learning models is that a wrong prediction from any of these models could lead to a wrong conclusion. Data fusion is a technique that merges various forms of data (obtained from different sensors or data sources) in order to procure more reconcilable and accurate information compared to the raw data that is mostly indecisive, imprecise, inconsistent, and conflicting [44]. Although, the PCR is the standard test for COVID-19 diagnosis, even this test is not completely accurate [45, 46]. Therefore, in this paper, the results of the three different deep learning models are combined together for COVID-19 CAD. The functionality of the utilized simple fusion method is illustrated in Fig. 8. As shown, a patient can take different tests (T_1, T_2, \dots, T_N) up to N tests, and each test will produce a result (R_1, R_2, \dots, R_N), 0 for negative and 1 for positive results. For every test, a weight will be assigned. Tests that have proven to be more accurate will have higher weights (e.g., PCR and X-ray), and cough and arrhythmia test results will have lower weights. These weights can be changed using the smart contract. After each result is received, they will be multiplied by their respective weight, and then a weighted average result will be calculated.

First of all, for each test, a corresponding ID will be assigned: [PCR: 1, Arrhythmia: 2, Cough: 3, CT/X-ray: 4]. Using these IDs, each result will be multiplied by the corresponding weight. The system will receive the results as:

$$[t_i, r_i] = \text{Result}\{(TR)_1, (TR)_2, \dots, (TR)_N\}, \tag{1}$$

where t stands for the test type, r is the result value of the respective test and can be either 0 or 1, and $(TR)_1, (TR)_2, \dots, (TR)_N$ are a series of tests with results. Result is a function that separates each test ID from its result. For example, if received a series is as $\{(1,0), (2,0), (3,1), (4,1)\}$, which means PCR and Arrhythmia are negative, but Cough and CT/X-ray tests were positive, t and r will be:

$$[t_1 = 1, r_1 = 0], [t_2 = 2, r_2 = 0], [t_3 = 3, r_3 = 1], [t_4 = 4, r_4 = 1], \tag{2}$$

and the weight of each test, previously defined and stored on the smart contract by either clinic operators (nurses or doctors) or developers, will be received as:

$$w_i = W(t_i), \tag{3}$$

where w is the corresponding weight. Finally, the final result FR will be calculated as:

Table 1 Corresponding weights of each test

Test	Weight
RT-PCR	1.2
Arrhythmia	0.4
Cough	0.8
CT/X-ray	2.6
CT/X-ray result approved by radiologist	3

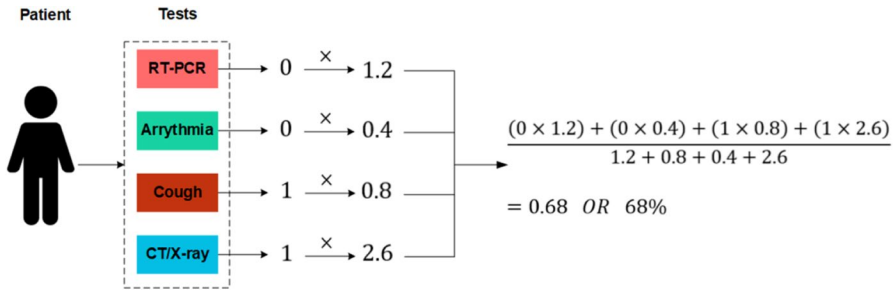


Fig. 9 A numerical example of the fusion method used in this paper

$$FR = \frac{\sum_{i=1}^N (r_i \times w_i)}{\sum_{i=1}^N (w_i)} \tag{4}$$

Table 1 shows the weights that specified in this research. Since chest X-ray and CT scan are reliable tools [9, 47] for COVID-19 diagnosis, higher weights are assigned to them. If the detection or segmentation is conducted via a deep learning model, the weight 2.6 will be assigned. However, if a doctor or radiologist approves the image as COVID-19 positive, the weight will be 3. The minimum summation of weights will be equal to 0, and the maximum will be either 5 (X-ray positive without approval) or 5.4 (CT/X-ray positive approved by a specialist).

Figure 9 shows a numerical example of the utilized diagnosis method. Here, RT-PCR and Arrhythmia test are negative, and Cough and X-ray are positive. Each result will be multiplied by its respective weight, and the sum of them will be divided by sum of weights, which results in 0.68 or a probability of 68% that the patient is COVID-19 positive. The decision process can be improved using various information fusion methods.

D-EI for Emergency Intelligence in COVID-19 Pandemic

In order to explore the applications of D-EI for emergency intelligence in pandemic conditions, first the machine learning based CAD results for COVID-19 are investigated.

Machine Learning Based CAD for COVID-19

Although COVID-19 mainly targets the respiratory system, major cardiac complications have also been reported [48]. If an emergency occurs and the patient is carried to a hospital, the data acquired in the ambulance is valuable. Therefore, the ambulance is also a critical entity or data holder in D-EI. This case study is implemented to investigate the applications of the deep learning-based model for predicting Arrhythmia on ECG classification using the D-EI proposed framework. For

Table 2 Classification results on MIT-BIH Arrhythmia dataset (test set)

Activity	Precision	Recall	F1-Score
N—Non-ectopic beats	0.85	0.99	0.92
S—Supraventricular ectopic beats	0.99	0.88	0.93
V—Ventricular ectopic beats	0.97	0.96	0.96
F—Fusion beats	0.98	0.94	0.96
Q—Unknown beats	1.00	0.99	1.00
Macro/weighted average	0.96	0.95	0.95

Table 3 Comparison of heartbeat classification using D-EI with state-of-the-art methods

Work	Average accuracy
This paper	95.2%
Kachuee et al. [50]	93.4%
Acharya et al. [51]	93.5%
Kaspal et al. [52]	93.24%
Malik et al. [53]	94.35%
Khan et al. [54]	95.2%
Alfaras et al. [55]	92.7%

this task, a publicly available dataset named MIT-BIH Arrhythmia Database [49] is used. The dataset, has five classes: ‘N’ or Non-ectopic beats (normal beat), ‘S’ or Supraventricular ectopic beats, ‘V’ or Ventricular ectopic beats, ‘F’ or Fusion Beats, and ‘Q’ or Unknown Beats.

The MIT-BIH Arrhythmia Database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings obtained from 47 subjects studied by the BIH Arrhythmia Laboratory. Twenty-three recordings were chosen at random from a set of 4000 24-h ambulatory ECG recordings collected from a mixed population of inpatients (about 60%) and outpatients (about 40%) at Boston’s Beth Israel Hospital; the remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that would not be well-represented in a small random sample. The recordings were digitized at 360 samples per second per channel with 11-bit resolution over a 10 mV range. Two or more cardiologists independently annotated each record; disagreements were resolved to obtain the computer-readable reference annotations for each beat (approximately 110,000 annotations in all) included with the database.

For implementation of the deep neural network model, Keras, was used with TensorFlow. Adam, a stochastic gradient descent method based on adaptive estimation of first-order and second-order moments, was used as the model’s optimizer, with values: $\text{learning}_{\text{rate}} = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. Batch sizes of 300, 400, and 500 were investigated with 50 and 100 epochs. The model trained with 100 epochs and a batch size of 400 yielded the best result. As the goal of this case study is to show the proposed framework in action. Therefore, we will not focus on the model’s details.

Table 4 An example of extracted feature from cough samples

Label	Chroma	Spectral_centroid	Zero_cross- ing_rate	mfcc1	...	mfcc20
COVID-19	0.412	2583.17	0.195	– 341.59	...	– 5.86
Normal	0.407	2710.82	0.142	– 346.86	...	– 0.5039

Table 5 Comparison of cough classification using D-EI with state-of-the-art methods

Work	Dataset	Average Accuracy
This paper	COVID-19 cough recordings [56]	100%
Imran et al. [10]	Their own COVID-19 cough dataset	95.6%
Sharma et al. [57]	Their own multi-class cough dataset [58]	66.74%
Vijayakumar et al. [59]	Their own COVID-19 cough dataset	94%
Mouawad et al. [60]	Their own COVID-19 cough dataset	99%

The model managed to classify different types of heartbeats with high accuracy. Since the ‘S’ and ‘N’ type beats are somewhat similar, 89 samples (11% of total samples) of ‘S’ type beats are wrongly predicted as ‘N’, but the rest of the results are acceptable. The best prediction performance belongs to ‘N’ and ‘Q’ beats. In D-EI, if an ECG sample is predicted to be ‘N’, the result sent to the system will be equal to 0. However, if ‘S’, ‘V’, ‘F’, and ‘Q’ classes are predicted, which means irregularity in heartbeat rhythm, the result sent to the system will be equal to 1. Precision, recall, *F1*-score, and support obtained by the model are shown in Table 2. As shown in the table, the average precision of these five classes is 0.96, and the average *F1*-score and recall are 0.95.

Finally, we compared our result with similar work such as [50], as shown in Table 3. As it can be seen, our approach, with larger batch size and epochs, performed better compared to other similar work. The studies mentioned in this table utilized the same MIT-BIH dataset in their proposed models.

The next investigated data is the COVID patients cough sounds. The dataset utilized for this case study [56] consists of a total of 170 cough samples, with 19 COVID-19 and 151 normal cough samples. First, every cough sample (audio file) was converted into a spectrogram, and after that, features such as Mel-frequency Cepstral coefficients (20 numbers in total), spectral centroid, zero-crossing rate, chroma frequencies, and spectral roll-off were extracted. Table 4 shows an example of this operation. These features are used for training the cough classification model.

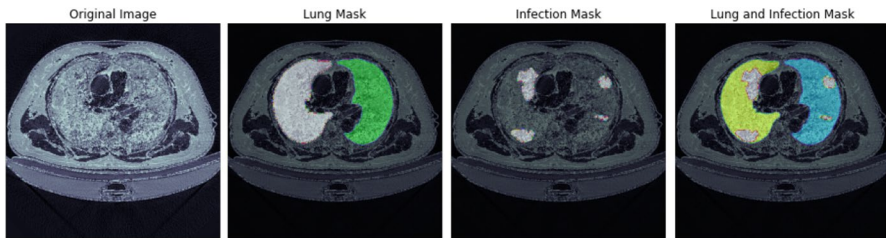
Adam and RMSProp optimizers are used for training different models and choosing the one with the best performance. The models were trained for 25 epochs, with batch sizes of 32, 64, and 128. The best model, with 100% training and validation accuracy, was the Adam trained model with a batch size of 128. Table 5 shows a comparison of the proposed model with recent COVID-19 related cough classification models.

Table 6 Classification report on X-ray dataset (test set)

Activity	Precision	Recall	F1-Score
Adam—COVID-19	0.99	0.95	0.97
Adam—Normal	0.92	0.86	0.89
Adam—Pneumonia	0.94	0.97	0.96
Macro/weighted average	0.95	0.93	0.94
RMSProp—COVID-19	1.00	0.92	0.96
RMSProp—Normal	0.88	0.91	0.90
RMSProp—Pneumonia	0.96	0.96	0.96
Macro/weighted average	0.95	0.93	0.94

Table 7 Comparison of X-ray image classification using D-EI with state-of-the-art methods

Work	Image classes	Average accuracy
This paper	COVID-19, Normal, Pneumonia	93.6%
Ozturk et al. [62]	COVID+, Pneumonia, No-findings	87.0%
Sethy et al. [64]	COVID positive vs. COVID negative	95.4%
Yoo et al. [63]	COVID-19 vs. Tuberculosis	95%
Panwar et al. [65]	COVID-19 vs. Other chest diseases	88.1%

**Fig. 10** A sample of the CT scan dataset

The next data used in this study is chest X-ray images that are used in the early diagnosis of COVID-19. The chest X-ray dataset utilized in this study [61] contains 6432 X-ray images belonging to three classes: COVID-19, normal, and pneumonia. By default, 80% of the dataset is set for training, and the remaining 20% for testing, with 460 COVID-19, 1,266 normal, and 3418 pneumonia samples in the training set, and 116, 317, and 855 samples for COVID-19, normal, pneumonia classes in the test set. For training the deep neural network model, the images were resized to 100 × 100 pixels, and data augmentation techniques, including width and height shift of 0.1, and shear and a zoom range of 0.1 were applied for training the model. The batch size of 64 is used, and models were trained for 50 epochs with Adam and RMSProp optimizers, yielding accuracies of 93.56% and 93.63%, respectively. The Adam trained model detected 14% of

Fig. 11 Predicted infection mask

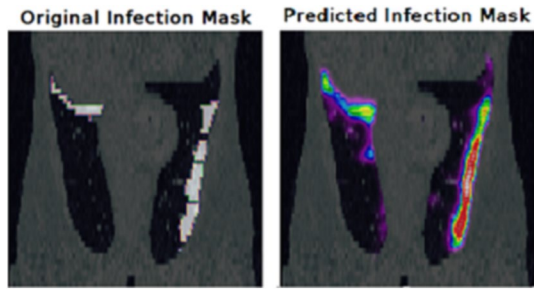


Table 8 Comparison of CT scan segmentation for COVID-19 using D-EI with state-of-the-art methods

Work	Average accuracy
This paper	99.61%
Singh et al. [70]	93.3%
Wang et al. [71]	90.1%
Ahuja et al. [72]	99.4%

normal test samples as pneumonia, and there was a 5% error in distinguishing COVID-19 test samples from pneumonia. Furthermore, the RMSprop trained model predicted COVID-19 test samples as normal and pneumonia with an error of 3% and 4% of respectively. Table 6 shows classification report of the Adam and RMSprop models. As shown, in the RMSprop trained model, the precision of COVID-19 is 1 (100%), however, the precision for normal class is 0.88 (88%) in this model.

Finally, Table 7 shows a comparison of recent studies focused on CAD using chest X-ray classification. The trained model for this paper outperformed [62], which used the same image classes as this paper, based on performance. Furthermore, [63] and [64] outperformed the average performance of this paper since they focused on a binary classification and used one less class, which can significantly impact the performance.

The next data used in this study are CT scans that act as a supportive tool in diagnosis of COVID-19. Deep learning trained models that can find evidence of COVID-19 in CT scans can act as a diagnosis tool, especially in case of a shortage of expert radiologists and doctors. For this task, a dataset containing 20 CT scans of COVID-19 diagnosed patients together with the segmentation of lungs and infections was utilized [66]. This dataset is a combination of [67, 68] and [69]. Figure 10 shows a sample of this dataset, which contains the original CT image, lung mask, infection mask, and a combination of both masks.

The U-net architecture was utilized for this task and 2 CT scans were used for the test set and the remaining 18 scans for the training set. This time, only Adam optimizer was utilized, and in order to prevent overfitting, the early stopping method on validation accuracy was utilized. The model was to be trained for 25 epochs; however, due to the usage of early stopping, it was stopped after 9 epochs, with a validation accuracy of 99.61%. Figure 11 shows a predicted infection mask by this model.

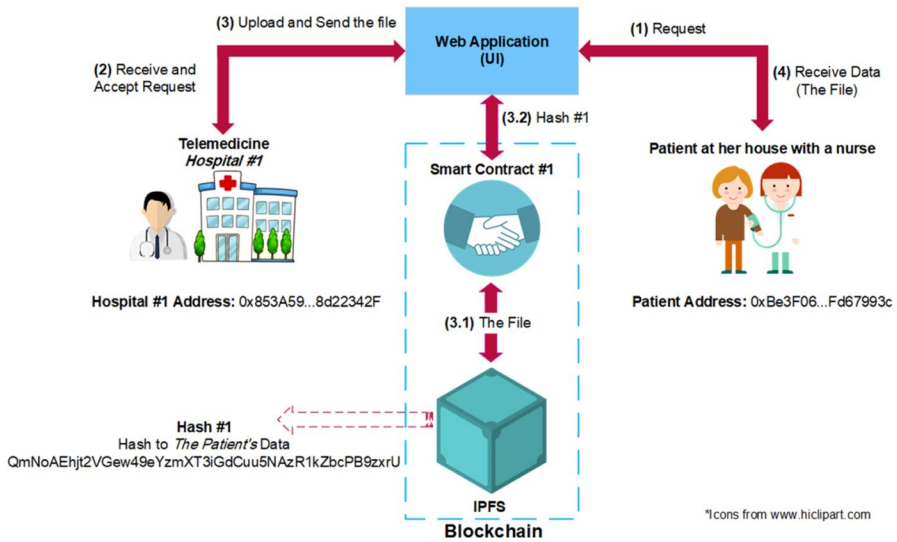


Fig. 12 D-EI framework for sharing data between different data holders for telemedicine

A comparison of recent papers focused on using CT scan segmentation for COVID-19 cases is presented in Table 8. As shown, this paper's model outperforms recent studies focused on this task based on accuracy.

D-EI for CAD as a Service on Decentralized Medical Cloud

Figure 12 shows the D-EI framework for sharing data and medical records between a telemedicine provider and a patient at home. This model is especially useful for COVID-19 patients with mild symptoms which are advised to stay at home.

In Fig. 12, the scenario is as follow: (1) The patient requests access to the records through *Smart Contract #1* (*Patient smart contract*—which is used for managing transactions between patients and medical centers). (2) After the authentication of the patient using her address on the blockchain and accepting the request, since the requested data is large and cannot be shared through the blockchain directly, *Data holder* (or *Telemedicine provider* in the figure) uploads the record on IPFS (3.1), and then receives a checksum hash (Hash #1–3.2). An advantage that IPFS provides is that no additional information, other than the hash, is required to retrieve the file. Moreover, IPFS does not allow another distinct file with the same hash to be created. Therefore, a file uploaded on IPFS can easily be verified. Next, again through the *SC#1*, this hash will be sent to *The Patient* on the blockchain on a secure channel. However, before that, *SC#1* checks if the hash is correct and if it exists on IPFS. If so, then the hash (*Hash #1*) will be passed to *The Patient*, and she can download the records using the IPFS hash address (4). As the figure shows, the smart contract and IPFS are located on the Blockchain, thus a secure channel is constructed between stakeholders. Note that in the figure, addresses are unique identifiers of each stakeholder.

When a CAD expert wants to train a diagnosis model on data from *Emergency/ Telemedicine provider*, first, the data holder is required to provide a sample of its dataset, along with descriptions of the dataset (Step 0). Next, a researcher connects to the *SC#2* (or *ModelTraining* contract, which is dedicated to handling contracts between data holders and researchers) via a web application (or UI). Then, in order to sign a contract with data holders and send a request for training his model on their dataset, the researcher must provide a proposal of what he intends to do with the data, which programming language, library, frameworks, and resources are needed in order to train his model (1). Moreover, at this stage the researcher is required to upload his code and algorithm on IPFS (*Train_Model.py* file—1.1). Then, he will share the hash (*Hash #2*—1.2) via *SC#2* and send his request. This smart contract will check if the file exists and the hash is correct. Then, it will be passed to *Hospital #1 (Data Holder)*, which if accepts the request (2), will be able to download the code, and start the training process (3). Otherwise, no access to the code will be given to anyone else, and the researcher will be notified that his request was declined. After the training process is over, the data holder will upload (4) the trained model file (*BestModel.h5*—4.1) or any other checkpoints and files if available, and share the hash (*Hash #3*—4.2), which the researcher will use to download the model (5). In this scenario, no data will be shared with any person, model owners, or researchers, thus the privacy of data holders will be preserved.

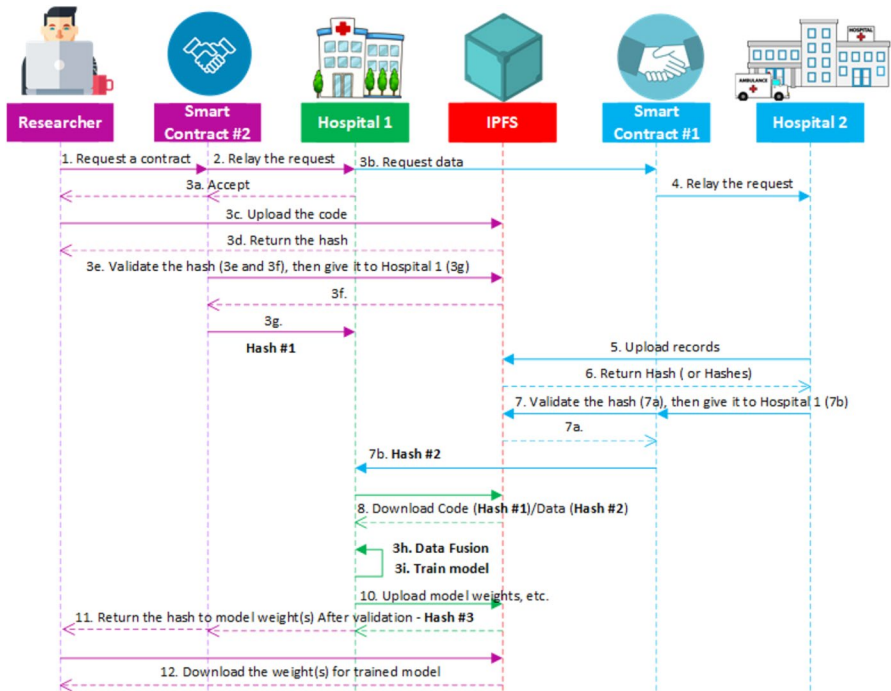


Fig. 13 Sequence diagram of the proposed framework for COVID CAD model training

The diagram for training CAD machine learning models using D-EI, is presented in Fig. 13. As mentioned, the datasets used in this study are publicly available. However, for demonstrating the application of D-EI we assume that they are not available. Moreover, let us assume that *Hospital #1* has 30 subjects, and *Hospital #2* has the remaining 17 subjects. The CAD researcher wants to train a model on this data. Therefore, he will sign a contract with *Hospital #1* using the smart contract (*SC#2*). The hospital accepts the request, however, as mentioned, it does not hold the complete records. Therefore, *Hospital #1* will try to request the remaining data from *Hospital #2* via *SC#1*. Then, *Hospital #2* will upload its data on IPFS, and share the hash. *SC#1* will check if the hash is valid, if so, *Hospital #1* will receive the hash (*Hash #2*) from *SC#1* and download the data (*mitbih.part1.csv* file—the file containing *Hospital #2*'s portion of the complete dataset) from IPFS with the hash. Meanwhile, the CAD researcher uploaded the code (*Train_Model.py*) on IPFS, and gave the hash (*Hash #1*) to *Hospital #1* so they could download the code. Then *Hospital #1* can start the training of the model. While the model is being trained, checkpoints will be saved periodically, and the hospital can share them upon request. When the process of training the model is finished, *Hospital #1* will upload the trained model's file (*h5* file), share the hash (*Hash #3*) via *SC#2*. Finally, the CAD expert can download the *h5* file using its hash.

In comparison with training the machine learning models in traditional centralized manner, the computational overhead of the proposed framework consists of two major parts. The first overhead is from sending and receiving the addresses of the files on IPFS and other interactions with the blockchain network. The second overhead is from execution of the smart contracts and validation of the hashes of various participants in the process. Since these two overheads are very small and the implementation of the D-EI framework as described in Sects. “Decentralized Deep Emergency Intelligence” and “D-EI for Emergency Intelligence in COVID-19 Pandemic” can be performed on an affordable computer system in real time. The main computational issue is training of the machine learning models, which like every machine learning training process, requires significant computational resources.

Conclusion

In this paper, a blockchain-based framework is proposed that allows collaboration of patients, healthcare providers and CAD researchers for access to medical data without concern for privacy. The proposed framework, can provide the machine learning based CAD systems with the required data for training accurate models. The application of the proposed D-EI framework is investigated on several COVID-19 diagnosis criteria and a machine learning based CAD system based on D-EI is presented. The patients ECG, CT scans, X-ray images and cough recordings are used for demonstration of a D-EI based COVID-19 CAD framework. It is shown that the diagnosis accuracy between 95%–100%, that is better than the reported in literature, has been achieved by the proposed CNN-based machine learning techniques using different input datasets. The experimental results show that the proposed framework

can provide an effective solution for rapid and privacy preserving CAD in pandemic conditions.

Funding This study has no funding.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Majidi, B., Hemmati, O., Baniardalan, F., Farahmand, H., Hajitabar, A., Sharafi, S., Aghajani, K., Esmaili, A., Manzuri, M.T.: Geo-spatiotemporal intelligence for smart agricultural and environmental eco-cyber-physical systems. In: *Enabling AI Applications in Data Science*, pp. 471–491. Springer (2021)
2. Nazerdeylami, A., Majidi, B., Movaghar, A.: Smart coastline environment management using deep detection of manmade pollution and hazards. In: *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, 2019. IEEE
3. Abbasi, M.H., Majidi, B., Eshghi, M., Abbasi, E.H.: Deep visual privacy preserving for internet of robotic things. In: *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, 2019. IEEE
4. Heldt, F.S., Vizcaychipi, M.P., Peacock, S., Cinelli, M., McLachlan, L., Andreotti, F., Jovanović, S., Dürichen, R., Lipunova, N., Fletcher, R.A., Hancock, A., McCarthy, A., Pointon, R.A., Brown, A., Eaton, J., Liddi, R., Mackillop, L., Tarassenko, L., Khan, R.T.: Early risk assessment for COVID-19 patients from emergency department data using machine learning. *Sci. Rep.* **11**(1), 4200 (2021). <https://doi.org/10.1038/s41598-021-83784-y>
5. Casiraghi, E., Malchiodi, D., Trucco, G., Frasca, M., Cappelletti, L., Fontana, T., Esposito, A.A., Avola, E., Jachetti, A., Reese, J., Rizzi, A., Robinson, P.N., Valentini, G.: Explainable machine learning for early assessment of COVID-19 risk prediction in emergency departments. *IEEE Access* **8**, 196299–196325 (2020). <https://doi.org/10.1109/ACCESS.2020.3034032>
6. Zame, W.R., Bica, I., Shen, C., Curth, A., Lee, H.-S., Bailey, S., Weatherall, J., Wright, D., Bretz, F., van der Schaar, M.: Machine learning for clinical trials in the era of COVID-19. *Stat. Biopharm. Res.* **12**(4), 506–517 (2020). <https://doi.org/10.1080/19466315.2020.1797867>
7. Gatto, M., Bertuzzo, E., Mari, L., Miccoli, S., Carraro, L., Casagrandi, R., Rinaldo, A.: Spread and dynamics of the COVID-19 epidemic in Italy: effects of emergency containment measures. *Proc. Natl. Acad. Sci.* **117**(19), 10484 (2020). <https://doi.org/10.1073/pnas.2004978117>
8. de Moraes Batista, A.F., Miraglia, J.L., Rizzi Donato, T.H., Porto Chiavegatto Filho, A.D.: COVID-19 diagnosis prediction in emergency care patients: a machine learning approach. *medRxiv* (2020). <https://doi.org/10.1101/2020.04.04.20052092>
9. Ducray, V., Vlachomitrou, A.S., Bouscambert-Duchamp, M., Si-Mohamed, S., Gouttard, S., Manuy, A., Wickert, F., Sigal, A., Gaymard, A., Talbot, F., Michel, C., Perpoint, T., Pialat, J.-B., Rouviere, O., Milot, L., Cotton, F., Douek, P., Rabilloud, M., Bousset, L., Argaud, L., Aubrun, F., Bohe, J., Bonnefoy, M., Chapurlat, R., Chassard, D., Chidiac, C., Chuzeville, M., Confavreux, C., Couraud, S., Devouassoux, G., Durieu, I., Fellahi, J.-L., Gaujard, S., Gaymard, A., Hot, A., Krolak-Salmon, P., Lantelme, P., Lina, B., Luaute, J., Lukaszewicz, A.C., Martin-Gaujard, G., Mornex, J.F., Potinet, V., Rimmelé, T., Rode, G., Sève, F.P., Sigal, A., Zoulim, F.: Chest CT for rapid triage of patients in multiple emergency departments during COVID-19 epidemic: experience report from

- a large French university hospital. *Eur. Radiol.* **31**(2), 795–803 (2021). <https://doi.org/10.1007/s00330-020-07154-4>
10. Imran, A., Posokhova, I., Qureshi, H.N., Masood, U., Riaz, M.S., Ali, K., John, C.N., Hussain, M.D.I., Nabeel, M.: AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app. *Informatics in Medicine Unlocked* (2020). <https://doi.org/10.1016/j.imu.2020.100378>
 11. Yu, K.P., Tan, L., Aloqaily, M., Yang, H., Jararweh, Y.: Blockchain-enhanced data sharing with traceable and direct revocation in IIoT. *IEEE Trans. Ind. Inf.* (2021). <https://doi.org/10.1109/TII.2021.3049141>
 12. Tan, L., Xiao, H., Yu, K., Aloqaily, M., Jararweh, Y.: A blockchain-empowered crowdsourcing system for 5G-enabled smart cities. *Comput. Standards & Interfaces* **76**, 103517 (2021). <https://doi.org/10.1016/j.csi.2021.103517>
 13. Gupta, M., Jain, R., Kumari, M., Narula, G.: Securing healthcare data by using blockchain. In: Namasudra, S., Deka, G.C. (eds.) *Applications of blockchain in healthcare*, pp. 93–114. Springer Singapore, Singapore (2021)
 14. Sharma, P., Jindal, R., Borah, M.D.: Healthify: a blockchain-based distributed application for health care. In: Namasudra, S., Deka, G.C. (eds.) *Applications of blockchain in healthcare*, pp. 171–198. Springer Singapore, Singapore (2021)
 15. Bittins, S., Kober, G., Margheri, A., Masi, M., Miladi, A., Sassone, V.: Healthcare data management by using blockchain technology. In: Namasudra, S., Deka, G.C. (eds.) *Applications of blockchain in healthcare*, pp. 1–27. Springer Singapore, Singapore (2021)
 16. Malina, L., Srivastava, G., Dzurenda, P., Hajny, J., Ricci, S.: A privacy-enhancing framework for internet of things services. In: *International Conference on Network and System Security*, pp. 77–97. Springer International Publishing, in *Network and System Security*, Cham (2019)
 17. Shen, M., Tang, X., Zhu, L., Du, X., Guizani, M.: Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities. *IEEE Internet Things J.* **6**(5), 7702–7712 (2019)
 18. Makhdoom, I., Zhou, I., Abolhasan, M., Lipman, J., Ni, W.: PrivySharing: a blockchain-based framework for privacy-preserving and secure data sharing in smart cities. *Comput. & Secur.* (2020). <https://doi.org/10.1016/j.cose.2019.101653>
 19. Chen, C.-L., Deng, Y.-Y., Weng, W., Sun, H., Zhou, M.: A blockchain-based secure inter-hospital EMR sharing system. *Appl. Sci.* **10**(14), 4958 (2020)
 20. Dai, H.-N., Imran, M., Haider, N.: Blockchain-enabled Internet of Medical Things to Combat COVID-19. *arXiv preprint arXiv:2008.09933* (2020)
 21. Shu, H., Qi, P., Huang, Y., Chen, F., Xie, D., Sun, L.: An efficient certificateless aggregate signature scheme for blockchain-based medical cyber physical systems. *Sensors* **20**(5), 1521 (2020)
 22. Jaleel, A., Mahmood, T., Hassan, M.A., Bano, G., Khurshid, S.K.: Towards medical data interoperability through collaboration of healthcare devices. *IEEE Access* **8**, 132302–132319 (2020)
 23. Yu, K., Tan, L., Shang, X., Huang, J., Srivastava, G., Chatterjee, P.: Efficient and privacy-preserving medical research support platform against COVID-19: a blockchain-based approach. *IEEE Consumer Electron. Magazine* **10**(2), 111–120 (2021). <https://doi.org/10.1109/MCE.2020.3035520>
 24. Fadaeddini, A., Majidi, B., Eshghi, M.: Secure decentralized peer-to-peer training of deep neural networks based on distributed ledger technology. *J. Supercomput.* **76**(12), 10354–10368 (2020)
 25. Fadaeddini, A., Majidi, B., Eshghi, M.: Privacy preserved decentralized deep learning: A blockchain based solution for secure ai-driven enterprise. In: *International Congress on High-Performance Computing and Big Data Analysis*. Springer (2019)
 26. Mohanta, B.K., Panda, S.S., Jena, D.: An overview of smart contract and use cases in blockchain technology. In: *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–4 (2018)
 27. Huang, Y., Bian, Y., Li, R., Zhao, J.L., Shi, P.: Smart contract security: a software lifecycle perspective. *IEEE Access* **7**, 150184–150202 (2019). <https://doi.org/10.1109/ACCESS.2019.2946988>
 28. Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., Kulatova, N., Rastogi, A., Sibut-Pinote, T., Swamy, N.: Formal verification of smart contracts: Short paper. In: *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, pp. 91–96 (2016)
 29. Tsankov, P., Dan, A., Drachslers-Cohen, D., Gervais, A., Buenzli, F., Vechev, M.: Securify: practical security analysis of smart contracts. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 67–82 (2018)

30. Kalra, S., Goel, S., Dhawan, M., Sharma, S.: ZEUS: analyzing safety of smart contracts. In: NDSS, pp. 1–12 (2018)
31. OpenZeppelin. <https://openzeppelin.com/contracts/>. Accessed 27 Feb 2021
32. SmartDec. <https://smartcontracts.smartdec.net/>. Accessed 27 Feb 2021
33. Alt L., Reitwießner, C.: SMT-based verification of solidity smart contracts. In: International Symposium on Leveraging Applications of Formal Methods, pp. 376–388. Springer (2018)
34. Benet, J.: InterPlanetary File System. <https://ipfs.io/>. Accessed 17 Dec 2020
35. Entriken, W.: Introduction to smart contracts. <https://ethereum.org/en/developers/docs/smart-contracts/>. Accessed 30 Nov 2020
36. MetaMask. <https://metamask.io/>. Accessed 17 Dec 2020
37. Solidity. <https://docs.soliditylang.org/en/v0.5.0/resources.html>. Accessed 17 Dec 2020
38. Truffle Suite. <https://www.trufflesuite.com/>. Accessed 17 Dec 2020
39. Ganache. <https://www.trufflesuite.com/ganache>. Accessed 17 Dec 2020
40. web3.js. <https://web3js.readthedocs.io/en/v1.3.0/>. Accessed 17 Dec 2020
41. OpenZeppelin's AccessControl Module. <https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/access/AccessControl.sol>. Accessed 2 Mar 2021
42. Kovan Testnet. <https://kovan-testnet.github.io/website/>. Accessed 15 Dec 2020
43. Infura. <https://infura.io/>. Accessed 15 Dec 2020
44. Meng, T., Jing, X., Yan, Z., Pedrycz, W.: A survey on machine learning for data fusion. *Inf. Fusion* **57**, 115–129 (2020). <https://doi.org/10.1016/j.inffus.2019.12.001>
45. Arevalo-Rodriguez, I., Buitrago-Garcia, D., Simancas-Racines, D., Zambrano-Achig, P., Del Campo, R., Ciapponi, A., Sued, O., Martinez-Garcia, L., Rutjes, A.W., Low, N.: False-negative results of initial RT-PCR assays for COVID-19: a systematic review. *PloS One* **15**(12), 0242958 (2020)
46. Watson, J., Whiting, P. F., Brush, J. E.: Interpreting a COVID-19 test result. *BMJ* **369** (2020)
47. Jain, R., Gupta, M., Taneja, S., Hemanth, D.J.: Deep learning based detection and analysis of COVID-19 on chest X-ray images. *Appl. Intell.* **51**(3), 1690–1700 (2021). <https://doi.org/10.1007/s10489-020-01902-1>
48. Shi, S., Qin, M., Shen, B., Cai, Y., Liu, T., Yang, F., Gong, W., Liu, X., Liang, J., Zhao, Q.: Association of cardiac injury with mortality in hospitalized patients with COVID-19 in Wuhan. *JAMA cardiology, China* (2020)
49. Moody, G.B., Mark, R.G.: The impact of the MIT-BIH Arrhythmia database. *IEEE Eng. Med. Biol. Mag.* **20**(3), 45–50 (2001)
50. Kachuee, M., Fazeli, S., Sarrafzadeh, M.: ECG Heartbeat Classification: a deep transferable representation. In: IEEE International Conference on Healthcare Informatics (ICHI) **2018**, 443–444 (2018)
51. Acharya, U.R., Oh, S.L., Hagiwara, Y., Tan, J.H., Adam, M., Gertych, A., San Tan, R.: A deep convolutional neural network model to classify heartbeats. *Comput. Biol. Med.* **89**, 389–396 (2017)
52. Kaspal, R., Alsadoon, A., Prasad, P.W.C., Al-Saiyd, N.A., Nguyen, T.Q.V., Pham, D.T.H.: A novel approach for early prediction of sudden cardiac death (SCD) using hybrid deep learning. *Multimedia Tools Appl.* (2020). <https://doi.org/10.1007/s11042-020-10150-x>
53. Malik, J., Loring, Z., Piccini, J.P., Wu, H.T.: Interpretable morphological features for efficient single-lead automatic ventricular ectopy detection. *J. Electrocardiol.* (2020). <https://doi.org/10.1016/j.jelectrocard.2020.11.014>
54. Khan, M.M.R., Siddique, M.A.B., Sakib, S., Aziz, A., Tanzeem, A.K., Hossain, Z.: Electrocardiogram heartbeat classification using convolutional neural networks for the detection of cardiac Arrhythmia. In: 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp. 915–920 (2020). <https://doi.org/10.1109/I-SMAC49090.2020.9243474>
55. Alfaras, M., Soriano, M.C., Ortín, S.: A fast machine learning model for ECG-based heartbeat classification and Arrhythmia detection, (in English). *Front. Phys.* (2019). <https://doi.org/10.3389/fphy.2019.00103>
56. COVID-19 Cough Recordings. <https://www.kaggle.com/himanshu007121/coughclassifier-trial>. Accessed 2 Mar 2021
57. Sharma, N., Krishnan, P., Kumar, R., Ramoji, S., Chetupalli, S.R., Ghosh, P.K., Ganapathy, S.: Coswara—A database of breathing, cough, and voice sounds for COVID-19 Diagnosis. *arXiv preprint arXiv:2005.10548* (2020)
58. Coswara dataset. <https://github.com/iiscleap/Coswara-Data>. Accessed 3 Mar 2021

59. Vijayakumar, D.S., Sneha, M.: Low cost Covid-19 preliminary diagnosis utilizing cough samples and keenly intellectual deep learning approaches. *Alexandria Eng. J.* **60**(1), 549–557 (2021). <https://doi.org/10.1016/j.aej.2020.09.032>
60. Mouawad, P., Dubnov, T., Dubnov, S.: Robust detection of COVID-19 in cough sounds. *SN Comput. Sci.* **2**(1), 34 (2021). <https://doi.org/10.1007/s42979-020-00422-6>
61. Chest X-ray (Covid-19 & Pneumonia). <https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia>. Accessed 2 Mar 2021
62. Ozturk, T., Talo, M., Yildirim, E.A., Baloglu, U.B., Yildirim, O., Acharya, U.R.: Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Comput. Biol. Med.* **121**, 103792 (2020)
63. Yoo, S.H., Geng, H., Chiu, T.L., Yu, S.K., Cho, D.C., Heo, J., Choi, M.S., Choi, I.H., Van Cung, C., Nhung, N.V.: Deep learning-based decision-tree classifier for COVID-19 diagnosis from chest X-ray imaging. *Front. Med.* **7**, 427 (2020)
64. Sethy, P.K., Behera, S.K., Ratha, P.K., Biswas, P.: Detection of coronavirus disease (COVID-19) based on deep features and support vector machine. *Preprints* (2020)
65. Panwar, H., Gupta, P.K., Siddiqui, M.K., Morales-Menendez, R., Singh, V.: Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVnet. *Chaos, Solitons & Fractals* **138**, 109944 (2020)
66. C.-C. Scans. <https://www.kaggle.com/andrewmvd/covid19-ct-scans>. Accessed 3 Mar 2021
67. Glick. COVID-19 Pneumonia. <https://radiopaedia.org/playlists/25887>. Accessed 3 Mar 2021
68. Paiva, O.: CT scans of patients with COVID-19 from Wenzhou Medical University. <https://coronacases.org/>. Accessed 3 Mar 2021
69. Jun, M., Cheng, G., Yixin, W., Xingle, A., Jiantao, G., Ziqi, Y., Mingqing, Z., Xin, L., Xueyuan, D., Shucheng, C., Hao, W., Sen, M., Xiaoyu, Y., Ziwei, N., Chen, L., Lu, T., Yuntao, Z., Qiongjie, Z., Guoqiang, D., Jian, H.: COVID-19 CT Lung and Infection Segmentation Dataset. <https://doi.org/10.5281/zenodo.3757475>. Accessed 3 Mar 2021
70. Singh, D., Kumar, V., Kaur, M.: Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks. *Eur. J. Clin. Microbiol. Infectious Dis.* **39**(7), 1379–1389 (2020)
71. Wang, X., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., Liu, W., Zheng, C.: A weakly-supervised framework for COVID-19 classification and lesion localization from chest CT. *IEEE Trans. Med. Imaging* **39**(8), 2615–2625 (2020)
72. Ahuja, S., Panigrahi, B.K., Dey, N., Rajinikanth, V., Gandhi, T.K.: Deep transfer learning-based automated detection of COVID-19 from lung CT scan slices. *Appl. Intell.* **51**(1), 571–585 (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Amirhossein Peyvandi¹ · Babak Majidi^{1,2}  · Soodeh Peyvandi³ · Jagdish Patra⁴

¹ Department of Computer Engineering, Khatam University, Tehran, Iran

² Emergency and Rapid Response Simulation (ADERSIM) Artificial Intelligence Group, Faculty of Liberal Arts and Professional Studies, York University, Toronto, Canada

³ Process Management and Business Intelligence, University of Applied Sciences Upper Austria, Steyr, Austria

⁴ Faculty of Science, Engineering and Technology, Swinburne University of Technology, Melbourne, Australia