Information Sciences 570 (2021) 124–143

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A novel framework of collaborative early warning for COVID-19 based on blockchain and smart contracts

Liwei Ouyang [a,b], Yong Yuan [c,d,f,*], Yumeng Cao [b], Fei-Yue Wang [b,e,f]

[a] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
[b] State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[c] School of Mathematics, Renmin University of China, Beijing 100872, China
[d] Engineering Research Center of Finance Computation and Digital Engineering, Ministry of Education, Beijing 100872, China
[e] Qingdao Academy of Intelligent Industries, Qingdao 266109, China
[f] Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China

ARTICLE INFO

Article history:
Received 29 December 2020
Received in revised form 8 March 2021
Accepted 2 April 2021
Available online 08 April 2021

Keywords:
Blockchain
Smart contracts
Federated learning
Learning markets
Collaborative early warning

ABSTRACT

Early warning is a vital component of emergency response systems for infectious diseases. However, most early warning systems are centralized and isolated, thus there are potential risks of single evidence bias and decision-making errors. In this paper, we tackle this issue via proposing a novel framework of collaborative early warning for COVID-19 based on blockchain and smart contracts, aiming to crowdsource early warning tasks to distributed channels including medical institutions, social organizations, and even individuals. Our framework supports two surveillance modes, namely, medical federation surveillance based on federated learning and social collaboration surveillance based on the learning markets approach, and fuses their monitoring results on emerging cases to alert. By using our framework, medical institutions are expected to obtain better federated surveillance models with privacy protection, and social participants without mutual trusts can also share verified surveillance resources such as data and models, and fuse their surveillance solutions. We implemented our proposed framework based on the Ethereum and IPFS platforms. Experimental results show that our framework has advantages of decentralized decision-making, fairness, auditability, and universality. It also has potential guidance and reference value for the early warning and prevention of unknown infectious diseases.

1. Introduction

Historically, the global outbreak and epidemic of acute infectious diseases have been witnessed to directly affect public health and socioeconomic development. Currently, the coronavirus disease 2019 (COVID-19) pandemic continues to spread globally. As of December 2020, it has already caused more than 70 million confirmed cases and one million deaths worldwide [58]. It is generally believed that the key to the prevention and control of infectious diseases is the long-term surveillance and rapid response of abnormal occurrences or increasing trends of infectious diseases by using early warning technologies and systems [24]. However, controlled by specific centralized organizations like national governments or Cen-

* Corresponding author at: School of Mathematics, Renmin University of China, Beijing 100872, China.
E-mail addresses: ouyangliwei2018@ia.ac.cn (L. Ouyang), yong.yuan@ia.ac.cn (Y. Yuan), yumeng.cao@ia.ac.cn (Y. Cao), feiyue.wang@ia.ac.cn (F.-Y. Wang).

https://doi.org/10.1016/j.ins.2021.04.021
0020-0255/© 2021 Elsevier Inc. All rights reserved.

ters for Disease Control and Prevention (CDC), most of the existing early warning systems (EWSs) for infectious diseases are passive and disease-specific, resulting in numerous isolated systems lacking reliable information sharing mechanisms. As such, they inevitably have the risks of single evidence bias and decision-making errors, and can thus only haphazardly pick up currently unknown diseases such as severe acute respiratory syndrome (SARS) and COVID-19 [34].

Introducing distributed collaboration into the EWSs can give full play to the crowd intelligence to break the resource and capacity limitations of the existing centralized systems. For example, influenza and Ebola surveillance based on crowdsourced public reporting data proves to be more proactive and timely than surveillance based on diagnostic data from medical institutions [47,28]. Since infectious disease surveillance and early warning is highly interdisciplinary, spanning a wide range of expertises, technologies, and data sources, it is widely believed that it is necessary to further integrate this collaboration of multiple contributors into the lifecycle of early warning. Therefore, there is an urgent need to establish a decentralized EWS to leverage the distributed contributions of multiple parties, support various early warning technologies, and monitor multi-source surveillance data. Furthermore, in an open, dynamic and complex environment, this EWS should be able to: 1) provide a secured and private collaboration environment for participants without mutual trust; 2) provide fair incentive mechanisms that promote sustained and effective contributions; 3) ensure the auditability, traceability, and credibility of surveillance data, models and early warning results to detect and eliminate malicious behaviors.

This research is targeted at filling in this important gap, and our contributions can be summarized as follows.

First, to break the resource constraints in centralized and semi-centralized systems, based on techniques including blockchain, smart contracts and artificial intelligence (AI), we propose a novel and decentralized framework of collaborative early warning for COVID-19 that meets the above requirements. Specifically, blockchain can help build a trustless collaborative environment, smart contracts can encapsulate and execute scalable warning rules and incentive mechanisms, and AI can enable a variety of pluggable surveillance models.

Second, to make full use of major contributing forces of medical institutions and social participants (i.e., organizations and individuals), we design two surveillance modes in a unified framework, i.e., medical federation surveillance based on federated learning and social collaboration surveillance based on learning markets. Therefore, multi-source data collected from medical channels and social channels with different data features and privacy sensitivity can be separately analyzed but collaboratively surveilled. Specifically, it is hoped that by using our framework, medical institutions can train better federated surveillance models with privacy protection, while social participants without mutual trusts can share verified surveillance resources such as data and models, and fuse their solutions. Accordingly, their combination can enhance the quality of decision-making and improve the performance of early warning.

Third, to evaluate our proposed framework, we explain our design in detail, compare it with the existing EWSs, and preliminarily implement and analyze it based on the Ethereum and Inter-Planetary File System (IPFS) platforms. An early warning scenario for COVID-19 based on open-source chest X-ray data sets is set up to comprehensively verify the functionalities of the framework, and the performance with multiple quantitative and qualitative indicators is discussed. Experiments show that the proposed framework can successfully combine the scattered medical and social surveillance forces, and provide them with fair incentives and trusted information sharing mechanisms, thereby reducing the risk of decision bias. Also, our framework with advantages of auditability and universality can further help the early warning and prevention of other unknown infectious diseases.

The remainder of this paper is organized as follows. Section 2 reviews the existing early warning technologies and systems, basic concepts of blockchain and smart contracts, as well as blockchain-based federated learning and learning markets; Section 3 elaborates our novel framework of collaborative early warning for COVID-19; Section 4 presents an illustrative implementation of our framework based on Ethereum and IPFS platforms and analyzes its performance; Section 5 discusses our future work, and Section 6 concludes.

2. Literature review

In this section, we briefly review the basic concepts and recent advances of existing early warning technologies and systems, blockchain and smart contracts, as well as blockchain-based federated learning and learning markets.

2.1. Early warning for infectious diseases

The major task of early warning for infectious diseases is to detect notable aberrations via analyzing surveillance data with specialized surveillance technologies, and on this basis, send out warning signals related to possible outbreaks before or at an early stage of the events to protect people from potential health risks [60]. According to the different surveillance data, the existing infectious disease EWSs can be divided into indicator-based and event-based [41]. Generally, indicator-based EWSs analyze structured data collected through routine surveillance channels including healthcare providers, diagnostic laboratories and governmental specialists, while event-based EWSs analyze unstructured data gathered from intelligence sources of any nature, such as search engine queries, social media posts, e-commerce sales trends, wearable device records and many other online big data generated outside the routine channels [18]. Therefore, the former is usually more reliable, while the latter is more timely. In order to obtain higher accuracy, most national EWSs, including the China Infectious Diseases Automated-Alert and Response System (CIDARS) [61], are indicator-based, while in order to obtain better timeliness, most web-based EWSs, such as the Influenzanet [19], HealthMap [14], ProMed-mail [42], and the Global Public Health Intelligence Network (GPHIN) [35], are event-based. Furthermore, according to different surveillance technologies, event-based EWSs can be divided into three main categories: news aggregators, automatic systems and human-moderated systems [27]. Specially, news aggregators include Influenzanet, because it only aggregates influenza activity index by collecting and screening influenza-like-illness (ILI) related questionnaires completed by volunteers, and users can not get more detailed information unless they examine each individual article [37]. Automatic systems go beyond the simple gathering task by adding a series of automatic analysis steps. HealthMap falls into this category because it uses natural language processing tools to automatically analyze Internet media reports and directly overlay warning results on interactive geographic maps [10]. Human-moderated systems additionally introduce a group of human analysts to screen for epidemiological relevance of the automatically processed information before presenting them to the users [6]. Both of ProMed-mail and GPHIN employ specialists to manually screen and aggregate their warning results to be published [66].

Obviously, the quality of surveillance data and the capability of surveillance technologies directly determine the performance of EWSs. More and more researches have showed that the utilization of multi-source surveillance data and the combination of various surveillance technologies can help promote active detection, reduce decision bias and balance the reliability and timeliness [60,45]. To achieve this integration, specific expertise provided by worldwide participants and their well-organized distributed decision-making are necessary. However, all of the aforementioned EWSs are centralized or semi-centralized with designated participants and fixed surveillance data and technologies, which means that any expansion of them might be expensive and inefficient. Therefore, we need to establish a new decentralized EWS to support such long-term and large-scale cooperation among participants without mutual trust. The blockchain and smart contracts technology, which can provide a high degree of interoperability, fair incentive mechanisms, trusted information sharing mechanisms, and secured and private collaboration environments, has become a very promising choice.

2.2. Blockchain and smart contracts

Originating from the widely publicized whitepaper *Bitcoin: A Peer-to-Peer Electronic Cash System* published by Satoshi Nakamoto in 2008 [36], blockchain is an append-only distributed ledger with chained data blocks maintained and shared by all nodes in a decentralized system [68]. According to the permission settings, blockchain can be classified into public blockchain, consortium blockchain, and private blockchain. Since blockchain is an integrated innovation based on the existing techniques, including cryptography, peer-to-peer networks, consensus algorithms, and incentive mechanisms, blockchain-based systems can achieve large-scale collaboration with privacy control by setting access permissions, as well as avoid data tampering and single point of failures through redundant storage [67]. Coined in 1994 by Nick Szabo [49] and revived by blockchain in recent years, smart contracts running on blockchain are self-enforced and self-verified computer programs with a series of pre-defined rules, which can be embedded into tangible or intangible assets, transactions and data, and then serve as a software-defined intermediary among untrusted participants to facilitate information exchange, value transfer, and asset management [54]. Since blockchain and smart contracts have such characteristics as trustlessness, autonomy, traceability, and tamper-resistance, they are widely considered as the perfect digital infrastructure for building distributed and disintermediary collaboration systems [39,52,26,70], and have been preliminarily applied in the field of public health with success [22], such as electronic medical records sharing [59,5], privacy data access control [2] and opioid prescription tracking [69].

This paper aims to establish a collaborative EWS framework for COVID-19 with the help of scalable and interoperable blockchain and smart contracts, so that our proposed EWS can not only improve the accuracy of early warning by promoting the sharing of trusted surveillance resources, the fusion of distributed decision-making, and the cooperation of untrusted contributors, but also improve the timeliness of early warning by adopting smart contracts with preset rules for real-time, automatic and continuous monitoring. Our implementation is based on two prevalent blockchain projects, i.e., Ethereum and IPFS. Ethereum is currently the most popular development platform for smart contracts [57], while IPFS is a peer-to-peer version-controlled distributed filesystem [3]. IPFS routes files according to their content hashes, aiming to share data globally without the risk of loss or tampering. In our design, the encrypted data and models to be shared are stored in IPFS, and only the obtained IPFS hashes are transmitted on the blockchain, thereby reducing the payload of data storage.

2.3. Blockchain-based federated learning and learning markets

Benefiting from its powerful data mining capabilities, the emerging AI technology has been witnessed to provide valuable solutions for many recent researches areas, including digital image processing [63,65], natural language processing [62,25] and multimodal analysis [16,64]. With the revolution of Internet and mobile devices, traditional single-factor surveillance technologies commonly used in indicator-based EWSs, such as CIDARS, gradually show their insufficient capacity in dealing with multi-source big data, and more and more AI-enabled intelligent models are introduced with better performance [12,38,56,51]. Considering that the surveillance data collected from medical channels and social channels has different data features and privacy sensitivity, our proposed framework is supposed to accordingly provide them with two different surveillance modes when introducing intelligent models, so as to promote their cross-end collaboration. Therefore, we combine and expand two existing works, i.e., federated learning (FL) [21] and learning markets (LM) [40].

FL is a new machine learning architecture proposed by Google in 2016 to solve the data islands problem. In FL, federation members jointly train a centralized federated model by exchanging and fusing local parameters, so as to implicitly aggregate training data and improve model performance without revealing private data. As such, FL is expected to help medical federations obtain a better federated model as their unified surveillance standard under privacy protection. However, there are two significant problems to be solved. First, the transmission and fusion of federated data based on a centralized cloud or server has potential risks of data loss, leakage, tampering, and single point of failures. Thus, Kim et al. propose a blockchainized FL architecture (BlockFL) for decentralization, in which federation members download all local updates stored in the latest block and update the global model locally [20]. Second, assuming that all members are trustworthy and reliable, FL indiscriminately accepts individual updates and shares the federated model, thus lacking the ability to exclude malicious behaviors and include fair and sustainable incentive mechanisms. Therefore, our previous work proposed a decentralized AI collaboration framework based on blockchain and smart contracts, namely, LM. LM consists of a collaboration market and a sharing market, and controls their logic through scalable market mechanisms encapsulated in programmable smart contracts. In the collaboration market, untrusted participants can realize distributed collaborative mining under dynamic incentives in ensemble learning (EL)'s "Centralized Data + Distributed Model (CDDM)" mode or FL's "Distributed Data + Centralized Model (DDCM)" mode. In the secured and private decentralized sharing market, auditable, traceable and trusted AI models and data can be traded as digital assets.

In [40], we take CDDM mode as an illustrative example to implement LM and prove its advantages. This paper expands LM to establish the system. Specifically, at the Medical End, we combine the ideas of [20] to expand the collaboration market, thereby realizing the FL's DDCM mode and completing the federated surveillance. At the Social End, we share trusted surveillance data and models in the sharing market, and complete collaborative surveillance in EL's CDDM mode in the collaboration market. It is worth noting that whether in LM or our proposed framework, smart contracts only manage the interaction logics between distributed participants, and the training of AI models is essentially completed off the blockchain and according to the existing methods of FL and EL, which means that we do not rely on specific models, and all intelligent surveillance models suitable for FL and EL can be easily integrated into our framework. Without loss of generality, three classic and widely studied deep learning models, including convolutional neural networks (CNN), long short-term memory (LSTM) networks and bi-directional long short-term memory (BiLSTM) networks, are selected to demonstrate and analyze the proposed framework [23,15,46,17].

3. Framework of collaborative early warning for COVID-19

In this section, we explain the framework, operation mechanisms, and detailed design of smart contracts of our proposed collaborative EWS for COVID-19.

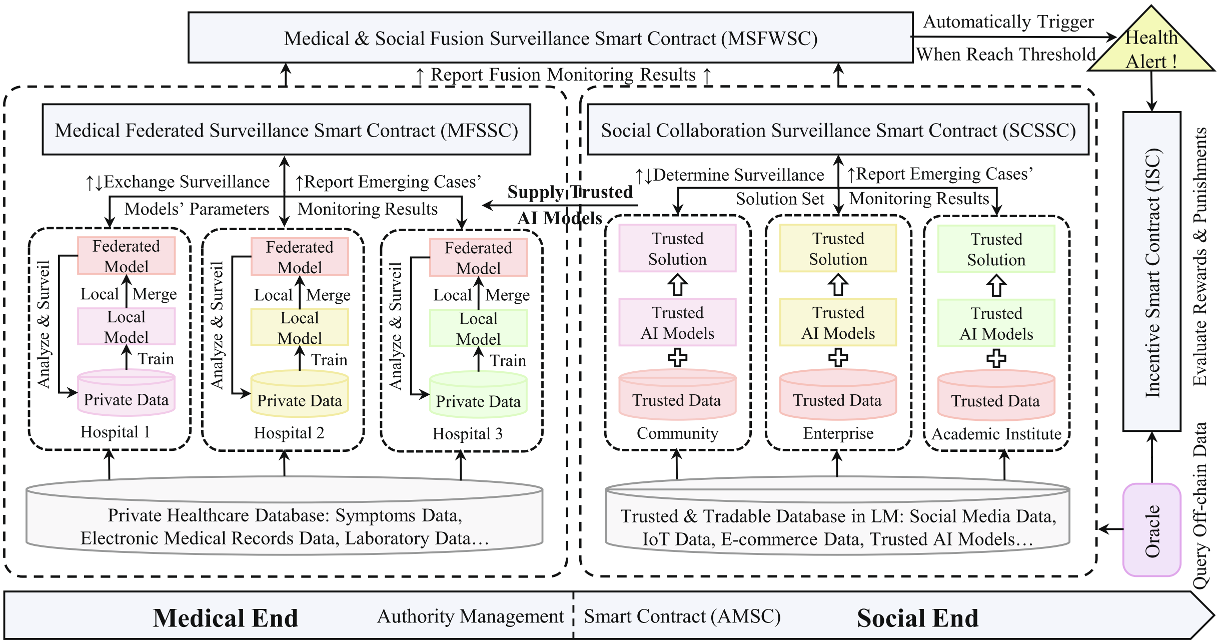

Medical & Social Fusion Surveillance Smart Contract (MSFWSC)
Automatically Trigger
When Reach Threshold
Health Alert !
↑ Report Fusion Monitoring Results ↑
Medical Federated Surveillance Smart Contract (MFSSC)
↑↓Exchange Surveillance Models' Parameters
↑Report Emerging Cases' Monitoring Results
Supply Trusted AI Models
Social Collaboration Surveillance Smart Contract (SCSSC)
↑↓Determine Surveillance Solution Set
↑Report Emerging Cases' Monitoring Results
Federated Model
Local
Merge
Local Model
Train
Private Data
Analyze & Surveil
Hospital 1
Hospital 2
Hospital 3
Trusted Solution
Trusted AI Models
Trusted Data
Community
Enterprise
Academic Institute
Private Healthcare Database: Symptoms Data, Electronic Medical Records Data, Laboratory Data…
Trusted & Tradable Database in LM: Social Media Data, IoT Data, E-commerce Data, Trusted AI Models…
Incentive Smart Contract (ISC)
Evaluate Rewards & Punishments
Oracle
Query Off-chain Data
Medical End
Authority Management Smart Contract (AMSC)
Social End

Fig. 1. The framework of proposed collaborative EWS for COVID-19.

3.1. An overview of the framework

The framework of the proposed system is shown in Fig. 1, and Table 1 presents the description of important notations and variables in this paper. The system running on an underlying blockchain network consists of Medical End and Social End, including four roles, i.e., medical federation members, social monitors, verifiers, and miners. Its main function is enabled by five smart contracts: Authority Management Smart Contract (AMSC), Medical Federated Surveillance Smart Contract (MFSSC), Social Collaboration Surveillance Smart Contract (SCSSC), Medical & Social Fusion Warning Smart Contract (MSFWSC), Incentive Smart Contract (ISC). At the Medical End, federation members realize federated surveillance based on FL and private healthcare data. At the Social End, social monitors and verifiers realize collaboration surveillance based on EL and trusted and tradable multi-source data in LM. At last, according to the preset fusion algorithms and early warning rules, MSFWSC fuses and examines emerging cases' monitoring results from two ends to realize automatic and real-time early warning of the new COVID-19 outbreak. In this process, the Oracle which links the off-chain verified authoritative websites is used as a trusted external data source for smart contracts to query the external states and trigger the execution. The main concepts in the system are defined as follows.

Participant: denoted by p, the participant who has registered in the blockchain network with assigned public key pk_p and private key sk_p.

Federation member: denoted by F, the participant from medical institutions who has private healthcare data for surveillance, and forms a federation $FList$ with other medical participants possessing the same type of data. F deposits to smart contracts before joining the early warning project, and negotiates to determine the FL mode and communication protocol in $FList$ before the project starts. The FL mode includes model types, fusion algorithms, iteration conditions, etc, and the communication protocol includes encryption/signature algorithms, shared key of private data k_{FList}, custom encrypted communication channel, etc.

Social monitor: denoted by S, the social participant who has verifiable data and models for surveillance, and forms a collaboration organization $SList$ with other social participants possessing the same type of data. S deposits to smart contracts before joining the early warning project, publishes surveillance solutions with corresponding data and models in LM, and collects sufficient proof of solution validity from verifiers. S can also publish surveillance solutions directly based on verified data and models in the sharing market.

Verifier: denoted by V, the social participant who is responsible for verifying the validity of surveillance data, models, and solutions in LM. V deposits to smart contracts before joining the early warning project. In the same round of validation, V can only testify for the same S once and cannot testify for himself. After the project ends, V will be rewarded according to the times of honest validations, and be punished when cheating.

Table 1
Notations and variables.

Notations and variables	Description
$p/F/S/V$	Participant/Federated members/Social monitors/Verifiers, $p \in \{F, S, V\}$
pk_p/sk_p	Public key/Private key of p
k_{S_j}/k_{FList}	Self-defined key of S_j or $FList$
$D_p/C_p/R_p$	Deposit/Credit/Reward of p
$FU_{A \to B}$	Fund transferred from A to B
$FList/SList/VList$	Confirmed list of $F/S/V$
$F_{dep}/S_{dep}/V_{dep}$	Total deposits of $FList/SList/VList$
$VSet_{F_i}/VSet$	Validation set from $F_i/FList$
$Model_{F_i}/Model_{S_j}$	Individual surveillance model of F_i/S_j
$Model_{FList_{F_i}}$	Federated surveillance model from F_i
$Model_{FList}$	Agreed federated surveillance model
E_{F_i}/Acc_{F_i}	Evaluation/Accuracy of $Model_{F_i}$ from F_i
E_{S_j}/Acc_{S_j}	Evaluation/Accuracy of $Model_{S_j}$ from S_j
$E_{FList_{F_i}}$	Evaluation of $Model_{FList_{F_i}}$ from F_i
E_{FList}/Acc_{FList}	Agreed evaluation/accuracy of $Model_{FList}$
EV_{S_jk}	Evaluation of $solution_{S_j}$ from V_k
EV_{S_jVList}	Agreed evaluation of $solution_{S_j}$
E_{SList}/Acc_{SList}	Estimated evaluation/accuracy of $\{solution_{S_j}\}$, $S_j \in sucSList$
$\alpha_{F_i}/\alpha_{S_j}$	Contribution coefficient of F_i/S_j
RJ_{F_i}/RF_{S_j}	Members rejected $Model_{F_i}/Model_{S_j}$
RW_{S_j}/RP_{S_j}	Members reviewed/accepted $Model_{S_j}$
$sucFList/puniFList$ $sucSList/puniSList$ $sucVList/puniVList$	The list of honest/dishonest $F/S/V$
$POS_{F_i}/POS_M/POS_S$	Amount of positive cases detected by F_i/Medical End/Social End
$PRED_{S_j}/PRED$	Prediction of new cases from $S_j/sucSList$
RES_M/RES_S	Medical/Social End's monitoring report

Miner: the participant who is responsible for the underlying consensus process, including verifying transactions, packing data blocks, and updating the blockchain. All participants can be miners.

AMSC: invoked for project and participant identity management. Project management includes setting project requirements and controlling the project process, and participant identity management includes adding/deleting p, updating p's information, and setting interaction permissions.

MFSSC: invoked by F for federated surveillance, including constructing unified validation sets, exchanging local surveillance models, fusing federated surveillance models, and monitoring emerging cases based on the obtained federated models.

SCSSC: invoked by S and V for collaboration surveillance, including sharing verifiable data and models in the sharing market, publishing individual surveillance solutions and performance in the collaboration market, verifying and evaluating individual surveillance solutions and performance, fusing the collaborative surveillance solution set, and monitoring emerging cases based on the obtained solution set.

MSFWSC: invoked for the automatic fusion and warning of emerging cases' monitoring results from Medical End and Social End, according to the preset fusion algorithms and early warning rules.

ISC: invoked for the calculation and implementation of the rewards and punishments of all participants, including cryptocurrency and credit scores.

3.2. Operation mechanisms

The proposed operation mechanisms can be divided into four stages, namely, project initialization, Medical End/Social End surveillance, M&S-fusion based warning, and post-warning incentivation. The detailed process in each stage is depicted as follows.

1. Project initialization:

   1) All participants register for project identity p with registration deposit D_{rg};

   2) Registered p applies to AMSC with the required fund $FU_{p \rightarrow AMSC}$ for becoming confirmed F in specific *FList*, confirmed S in specific *SList*, or confirmed V in *VList*;

   3) Confirmed F, S, or V can apply to AMSC for deleting role and exiting project before the project starts;

   4) When the amounts of F, S, and V meet the preset thresholds, the project starts.

2. Medical End/social End surveillance:

   A. Medical End surveillance

   1) $F_i \in FList$ shares validation set $VSet_{F_i}$ sampled from privately-held data set in *FList*, downloads and combines all $VSet_{F_i}$ to construct a unified validation set *VSet* locally;

   2) In each iteration, $F_i \in FList$ shares their local model $Model_{F_i}$ trained on privately-held data set in *FList* and publishes its performance E_{F_i} on *VSet* in the blockchain network. MFSSC calculates and publishes F_i's contribution coefficient α_{F_i};

   3) $F_i \in FList$ verifies and evaluates local models from others, excludes false models, fuses the federated model $Model_{FList_{F_i}}$ according to α_{F_i}, and at last, publishes $Model_{FList_{F_i}}$'s performance $E_{FList_{F_i}}$ on *VSet* and the list of members he/she has rejected;

   4) MFSSC determines the final federated model $Model_{FList}$ from the consensus of all $E_{FList_{F_i}}$ and publishes $Model_{FList}$'s performance E_{FList} on *VSet*, all α_{F_i}s, and the lists of rewards and punishments, i.e., *sucFList* and *puniFList*. This round of iteration ends and the next starts. Until the preset stop conditions are met, the training of federated surveillance is completed;

   5) After the training completes, all $F_i \in sucFList$ use the obtained federated model as the unified standard to monitor their received emerging cases. MFSSC fuses their uploaded results according to the preset rules and reports to MSFWSC.

   B. Social End surveillance.

   1) Social participants share verifiable surveillance data and models in the sharing market, and prepare for the subsequent collaboration;

   2) In each iteration, $S_j \in SList$ publishes surveillance solution $solution_{S_j}$ with corresponding verifiable data and models in LM, announces their performance E_{S_j} in the blockchain network, and requests sufficient $V_k \in VList$ to prove their validity;

   3) V_k verifies the validity and performance of $solution_{S_j}$ according to the preset validation rules and publishes the evaluation EV_{S_jk};

   4) SCSSC determines the final evaluation EV_{S_jVList} of $solution_{S_j}$ from the consensus of all EV_{S_jk}, publishes the lists of rewards and punishments, i.e., $sucSList, puniSList, sucVList, puniVList$, calculates all α_{S_j}s, and the weighted estimated performance E_{SList} of the final collaborative surveillance solution set $\{solution_{S_j}\}, S_j \in sucSList$. This round of iteration ends and the next starts. Until the preset stop conditions are met, the training of collaboration surveillance is completed;

5) After the training completes, all $S_j \in sucSList$ use their verified solutions in $\{solution_{S_j}\}$ to monitor their received emerging cases respectively. SCSSC fuses their uploaded results according to the preset rules and reports to MSFWSC.

3. M&S-Fusion based warning:
MSFWSC receives the monitoring reports RES_M from MFSSC and RES_S from SCSSC, fuses their monitoring results of emerging cases based on the preset algorithms and weights, i.e., W_M and W_S, and automatically alerts when the early warning conditions are met.

4. Post-warning incentivation:
ISC calculates and implements the rewards and punishments of all participants based on the lists of rewards and punishments from MFSSC and SCSSC and the comparison of system warning results with outbreak information from authoritative websites. The early warning project ends.

3.3. Design of smart contracts

The proposed collaborative EWS is realized by five smart contracts, i.e., AMSC, MFSSC, SCSSC, MSFWSC, and ISC, which are developed to inherit and monitor each other for simplicity and security. The preset project parameters and their definitions are shown in Table 2. Note that this section is only a simple illustrative implementation for the collaboration of one medical federation and one social collaboration organization. By adjusting the contract logics and corresponding parameters, it is easy to extend our participants, authorities, incentive mechanisms, fusion algorithms, and early warning rules. The details are as follows.

Fig. 2 shows a complete cycle of the EWS and the scope of smart contracts in the cycle. To distinguish with the four stages at two ends, seven bool-type status flags are set, i.e., $startPro, endMTrain, endSTrain, MFini, SFini, MSFini$, and $endPro$. As shown in Fig. 2, when these status flags are set as true, the next stage of interaction starts.

Algorithm 1. Authority Management Smart Contract.

```
Input: $startPro = False$; $D_{rg}$, $D_{rq}$, $FU_{p\rightarrow AMSC}$, $F_{lim}$, $S_{lim}$, $V_{lim}$, $role$;
Output: $startPro = True$; $FList$, $SList$, $VList$, $F_{dep}$, $S_{dep}$, $V_{dep}$;
1: $p$ deposits $D_{rg}$ to AMSC and registers in project, $p \leftarrow \{D_p : D_{rg}, C_p : 0\}$;
2: Before project starts, registered $p$ applies to be $F/S/V$ by executing $checkQualifications(p, role)$ to check:
1) apply with required fund $FU_{p\rightarrow AMSC} \geqslant D_{rq} - D_p$; 2) $len(FList/SList/VList) \leqslant F_{lim}/S_{lim}/V_{lim}$. If returns $True$,
    $D_p = D_p + FU_{p\rightarrow AMSC}$, when $role = 1$, $FList = FList \cup p$, $F_{dep} = F_{dep} + D_p$; when $role = 2$, $SList = SList \cup p$,
    $S_{dep} = S_{dep} + D_p$; when $role = 3$, $VList = VList \cup p$, $V_{dep} = V_{dep} + D_p$; else throws;
3: Before project starts, confirmed $F/S/V$ applies to delete $role$ and exit project:
    when $role = 1$, $FList = FList \setminus p$, $F_{dep} = F_{dep} - D_p$; when $role = 2$, $SList = SList \setminus p$, $S_{dep} = S_{dep} - D_p$; when $role = 3$,
    $VList = VList \setminus p$, $V_{dep} = V_{dep} - D_p$; at last, AMSC returns $FU_{AMSC\rightarrow p} = D_p$, $D_p = 0$;
4: If the length of $FList$, $SList$, and $VList$ are equal to $F_{lim}$, $S_{lim}$, and $V_{lim}$, respectively, project starts, sets
    $startPro = True$, and goes to step 5; else goes to step 2 and waits for more $F/S/V$;
5: return $FList$, $SList$, $VList$, $F_{dep}$, $S_{dep}$, $V_{dep}$;
```

AMSC: As shown in Algorithm 1, AMSC realizes the main functions in the project initialization stage, escrows all deposits and bonuses, and is monitored and invoked by other smart contracts to query and update project and participant information. Initial participant information includes deposit and credit, recorded as $p[account] = \{D_p, C_p\}$, and initial C_p can be set to be the same or proportional to initial D_p. After selecting roles and passing the validation of qualification function $checkQualifications(p, role), p$ joins $FList, SList$ or $VList$ and obtains the corresponding permissions. $checkQualifications(p, role)$ can be customized to check whether D_p and C_p meet preset thresholds. We set the simplest incentive mechanisms as follows: for cryptocurrency, all honest Fs and Ss divide F_{dep} and S_{dep} respectively according to their α_{F_i} and α_{S_j}, all Vs divide V_{dep} accord-

Table 2
The preset project parameters.

Notations and variables	Description
C_{p+}/C_{p-}	Single credit reward/punishment of p
$D_{rg}/D_{rq}/C_{rw}$	Required deposit/credit for registration/selecting role/reviewing solution
$F_{lim}/S_{lim}/V_{lim}$	Maximum amount of $F/S/V$
$\Theta_{rw}/\Theta_{E_S}/\Theta_{POS_S}/\Theta_A$	Thresholds of reviewed verifiers/accepting $Model_{S_j}$/determining a positive case/alert
W_M/W_S	Fusion weight of Medical End/Social End

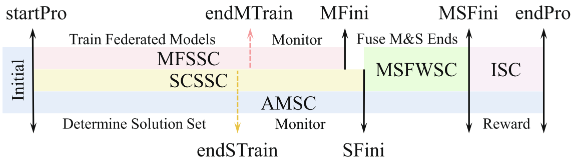


Fig. 2. A complete early warning cycle.

ing to their times of honest validations, and all fraudulent *F*s, *S*s, and *V*s lose their deposits. At the same time, for credit scores, honest *F*s, *S*s, and *V*s gain scores, while fraudulent *F*s, *S*s, and *V*s lose scores. More specific incentive mechanisms will be detailed in the discussion of ISC and Section 4.2.3.

Algorithm 2. Medical Federated Surveillance Smart Contract.

```
Input: ${startPro, endMTrain, MFini} = {True, False, False}$; $FList$;
Output: ${startPro, endMTrain, MFini} = {True, True, True}$; $\alpha_{F_i}$, $sucFList$, $puniFList$, $RES_M$;
1: All $F_i \in FList$ construct a unified $VSet$:
for each $F_i$: 1) encrypts $VSet_{F_i}$ with $k_{FList}$, 2) uploads encrypted $VSet_{F_i}$ on IPFS, and publishes the obtained hash
    $IPFS\{Enc\{VSet_{F_i}\}_{k_{FList}}\}$, 3) downloads all other $VSet_{F_i}$ to form $VSet$ locally;
2: All $F_i \in FList$ train $Model_{F_i}$ on private data sets and publish $E_{F_i}$ and $IPFS\{Enc\{Model_{F_i}\}_{k_{FList}}\}$. MFSSC computes
    $\alpha_{F_i} = getContri(E_{F_i})$;
3: All $F_i \in FList$ examine other local models, fuse federated model $Model_{FList_{F_i}}$ excluding false models and publish
    $E_{FList_{F_i}}$. If $F_i$ rejects $F_j$, $RJ_{F_j}.push(F_i)$. When all members finish, goes to step 4;
4: MFSSC obtains the consensus of $Model_{FList}$ and determines the final $sucFList$ and $puniFList$: 1) finds consistent
    $E_{FList}$ to determine the agreed $Model_{FList}$, 2) for each $F_i \in FList$: if $E_{FList_{F_i}} \neq E_{FList}$ or $len(RJ_{F_i}) \geqslant (len(FList) - 1)/2$,
    $puniFList.push(F_i)$, $\alpha_{F_i} = 0$; else $sucFList.push(F_i)$. When finishes, sets $endMTrain = True$, goes to step 5;
5: If $\{endMTrain, MFini\} = \{True, False\}$, all $F_i \in sucFList$ work together to monitor emerging cases based on
    $Model_{FList}$: for each $F_i$: 1) uploads the amount of positive cases $POS_{F_i}$, 2) MFSSC counts the sum
    $POS_M = \sum_{sucFList} POS_{F_i}$, and publishes the final report $RES_M = \{E_{FList}, POS_M\}$ to MSFWSC, when finishes, sets
    $MFini = True$, goes to step 6;
6: return $\alpha_{F_i}$, $sucFList$, $puniFList$, $RES_M$;
```

MFSSC: As shown in Algorithm 2, MFSSC realizes the main functions of Medical End surveillance in the Medical End/Social End surveillance stage, monitors AMSC to synchronizes permissions, and invokes MSFWSC to report medical monitoring results. Federated surveillance consists of two stages: training and monitoring. Before training starts, every $F_i \in FList$ needs to prepare a $VSet_{F_i}$ in their acceptable sample proportion to form a unified validation set *VSet* together, and subsequently take models' performance on *VSet* as the evaluation standard, so as to reduce the impact of private data bias, exclude overfitting local models and maintain the training fairness. Since *F* are motivated to provide models that outperforms on *VSet*, the obtained federated model will cater to members providing more private data, thereby encouraging moderate sharing. Federations can also be built based on certain trust, but technically, both of the unified *VSet* and trust basis are not mandatory but for better security, privacy, and performance. Because FL adopts the homogeneous basic models, *F*s only need to share, verify, and fuse the models' weight parameters as formula (1).

$$Weight(Model_{FList}) = \sum_{F_i \in sucFList} \alpha_{F_i} Weight(Model_{F_i}). \tag{1}$$

To record the interactive history and ensure security and privacy, the private data transmission protocol stipulated in our design is as follows: 1) all *F*s negotiate to determine the shared key k_{FList}, and every *S* generates his own symmetric key k_{S_j}, 2) *F* and *S* upload private files encrypted by k_{FList} or k_{S_j} on IPFS, 3) *F* and *S* publish the obtained IPFS hashes in the blockchain network, or transmit them to others by invoking the **Send** function in formula (2), *S* also sends the encrypted k_{S_j} to receivers, 4) receivers download and decrypt encrypted files to obtain original files. The evaluation of models can include a variety of indicators such as accuracy, recall rate, and F1 measure. $getContri(E_{F_i})$ is the calculation function of α_{F_i} that can adopt various forms. For instance, when E_{F_i} is regarded as Acc_{F_i}, it can be set inspired by AdaBoost [11], i.e., $\alpha_{F_i} = \frac{1}{2} \ln \frac{1-e_{F_i}}{e_{F_i}}$, where e_{F_i} is the weighted classification error rate of $Model_{F_i}$ on *VSet*. There is no special verifier in federated surveillance, and members are required to supervise, verify, and evaluate each other. A successful model needs the support of more than half of the other members, and the specific support ratios and consensus mechanisms can be easily extended in the future.

$$Send(from, to, Mes) = \{Msg, sig\{Msg\}_{sk_{from}}\}_{from \rightarrow to}, Msg = Enc\{Mes\}_{pk_{to}}. \tag{2}$$

Algorithm 3. Social Collaboration Surveillance Smart Contract.

```
Input: $\{startPro, endSTrain, SFini\} = \{True, False, False\}$; $SList$, $VList$, $k_{S_j}$, $C_{rw}$, $C_{V+}$, $C_{V-}$;
Output: $\{startPro, endSTrain, SFini\} = \{True, True, True\}$; $C_{S_j}$, $C_{V_k}$, $sucSList$, $puniSList$, $sucVList$, $puniVList$, $RES_S$;
1: Social participates share trusted data or models to obtain adequate credit scores for subsequent operations;
2: For the same data set, every $S_j \in SList$: 1) publishes encrypted $solution_{S_j}$'s hash, $IPFS\{Enc\{Model_{S_j}\}_{k_{S_j}}\}$, and $E_{S_j}$, 2)
   calls $Send(S_j, V_k, k_{S_j})$ to send $k_{S_j}$ to sufficient honest verifiers for validation;
3: $V_k \in VList \setminus (puniVList \cup RW_{S_j})$ & $S_j \neq V_k$ & $C_{V_k} > C_{rw}$ verifies $Model_{S_j}$ and publishes evaluation $EV_{S_jk}$: 1) every
   review costs credit scores $C_{rw}$, $C_{V_k} = C_{V_k} - C_{rw}$, $RW_{S_j}.push(V_k)$, 2) if $Model_{S_j}$ is invalid, $RF_{S_j}.push(V_k)$, else
   $RP_{S_j}.push(V_k)$; when $len(RW_{S_j}) = \Theta_{rw}$, goes to step 4;
4: SCSSC obtains the consensus evaluation of every $Model_{S_j}$, determines the collaborative solution set and the
   $sucSList$, $puniSList$ and $sucVList$, $puniVList$:
 1) determines the validity of $Model_{S_j}$:
   if $len(RP_{S_j}) \leqslant len(RF_{S_j})$, $Model_{S_j}$ is invalid, $puniSList.push(S_j)$, clears $E_{S_j}$, $\alpha_{S_j} = 0$, for each $V_k \in RP_{S_j}$,
   $puniVList.push(V_k)$, $C_{V_k} = C_{V_k} - C_{V-}$, for each $V_k \in RF_{S_j}$, $sucVList.push(V_k)$, $C_{V_k} = C_{V_k} + C_{V+}$, goes to step 4); else
   $Model_{S_j}$ is valid, goes to step 2),
 2) finds consistent $EV_{S_jVList}$ from all $EV_{S_jk}$:
   for each $V_k \in RW_{S_j}$, if $EV_{S_jk} = EV_{S_jVList}$ & $V_k \in RP_{S_j}$, $sucVList.push(V_k)$, $C_{V_k} = C_{V_k} + C_{V+}$, else $puniVList.push(V_k)$,
   $C_{V_k} = C_{V_k} - C_{V-}$,
 3) compares $E_{S_j}$ with $EV_{S_jVList}$, if $E_{S_j} = EV_{S_jVList}$ and $E_{S_j} > \theta_{E_S}$, $sucSList.push(S_j)$, $\alpha_{S_j} = getContri(E_{S_j})$, else
   $puniSList.push(S_j)$, $\alpha_{S_j} = 0$;
 4) if $len(sucSList) + len(puniSList) = len(SList)$, sets $endSTrain = True$, SCSSC computes estimated $E_{SList}$, goes to step
   5; else returns;
5: If $\{endSTrain, SFini\} = \{True, False\}$, all $S_j \in sucSList$ work together to monitor emerging cases through a
   weighted vote: for each $S_j \in sucSList$: 1) uploads $PRED_{S_j}$, 2) MFSSC computes $PRED$, $POS_S = Count(PRED \geqslant \theta_{POS_S})$,
   and publishes the final report $RES_S = \{E_{SList}, POS_S\}$ to MSFWSC, when finishes, $SFini = True$, goes to step 6;
6: return $C_{S_j}$, $C_{V_k}$, $sucSList$, $puniSList$, $sucVList$, $puniVList$, $RES_S$
```

SCSSC: As shown in Algorithm 3, SCSSC realizes the main functions of Social End surveillance in the Medical End/Social End surveillance stage, monitors AMSC to synchronizes permissions, and invokes MSFWSC to report social monitoring results. Collaboration surveillance is realized based on our previous work LM [40]. In LM, registered participants can share or verify surveillance data and models to obtain enough credit scores for subsequent operations, and S should upload or link the corresponding data and models when publishing solutions. Since LM with credit management traces the interaction behaviors of registered participants and the historical information of surveillance data and models in the whole process, S and V without mutual trust can choose collaborators, data, and models based on these records. Besides, medical participants can also request verified surveillance resources in the sharing market to reduce the difficulty of data analysis.

Collaboration surveillance also consists of two stages: training and monitoring. S in the same $SList$ can submit solutions for the same type of data. And when their data are not the same, similar to federated surveillance, they also need to construct a unified validation set for fairness before training and validation. Thus, for simplicity, Algorithm 3 assumes that they monitor the same trusted open-source data. V needs to consume credit scores when reviewing and evaluating solutions to increase the cost of invalid and malicious validation. S who upload invalid models or false performance will be punished, and V will be rewarded and punished according to the times of honest and dishonest validations. Only S who has been verified at least θ_{rw} times and passed can participate in the final monitoring, i.e., $sucSList.push(S_j)$. They will monitor emerging cases based on their verified solutions and weighted vote for the prediction results based on EL methods, and thus E_{SList} and $PRED$ are calculated as formula (3). The calculation function of α_{S_j} can be customized or the same as that of federated surveillance.

$$E_{SList} = \sum_{S_j \in sucSList} \left(\frac{\alpha_{S_j}}{\sum\limits_{S_j \in sucSList} \alpha_{S_j}} E_{S_j} \right), \quad PRED = \sum_{S_j \in sucSList} \left(\frac{\alpha_{S_j}}{\sum\limits_{S_j \in sucSList} \alpha_{S_j}} PRED_{S_j} \right). \tag{3}$$

Algorithm 4. Medical & Social Fusion Warning Smart Contract + Incentive Smart Contract

```
Input: $\{MFini, SFini, MSFini, endPro\} = \{True, True, False, False\}$; $W_M$, $W_S$, $RES_M$, $RES_S$; $FList$, $SList$, $VList$, $F_{dep}$, $S_{dep}$, $V_{dep}$, $C_{F+}$, $C_{F-}$, $C_{S+}$, $C_{S-}$; $C_{F_i}$, $\alpha_{F_i}$, $sucFList$, $puniFList$; $C_{S_j}$, $C_{V_k}$, $\alpha_{S_j}$, $sucSList$, $puniSList$, $sucVList$, $puniVList$;
Output: $\{MFini, SFini, MSFini, endPro\} = \{True, True, True, True\}$; $R_{F_i}$, $C_{F_i}$, $R_{S_j}$, $C_{S_j}$, $R_{V_k}$, $C_{V_k}$;
1: MSFWSC fuses the monitoring results $RES_{MS}$ of all $FList$s and $SList$s to alert, if $RES_{MS} \geqslant \Theta_A$, the system alerts;
   else the system does not alert, when finishes, $MSFini = True$, goes to step 2;
2: If $\{MSFini, endPro\} = \{True, False\}$, ISC compares warning results with authoritative websites, computes and implements participates' rewards and credits: $R_{F_i}, C_{F_i}, R_{S_j}, C_{S_j}, R_{V_k}, C_{V_k} = settleReward(FList, SList, VList, F_{dep}, S_{dep}, V_{dep}, C_{F+}, C_{F-}, C_{S+}, C_{S-}, C_{F_i}, \alpha_{F_i}, sucFList, puniFList, C_{S_j}, C_{V_k}, \alpha_{S_j}, sucSList, puniSList, sucVList, puniVList)$, when finishes, sets $endPro = True$, goes to step 3;
3: return $R_{F_i}$, $C_{F_i}$, $R_{S_j}$, $C_{S_j}$, $R_{V_k}$, $C_{V_k}$;
```

MSFWSC: As shown in Algorithm 4, MSFWSC realizes the main functions in the M&S-fusion based warning stage, monitors MFSSC and SCSSC to fuse their monitoring results and automatically alerts when the fusion results exceed the preset threshold. Specifically, RES_{MS} is fused as formula (4), where m and n are the amounts of $FList$s and $SList$s. Through monitoring multiple MFSSCs and SCSSCs, MSFWSC can fuse monitoring results from multiple medical federations and social collaboration organizations. It is generally believed that the medical surveillances are more accurate, while the social surveillances are more timely. Therefore, by adjusting the fusion weights of two ends, it has the potential to enhance the accuracy and timeliness of EWS at the same time. We will discuss this in detail in Section 4.2.1.

$$RES_{MS} = W_M \sum_m (E_{FList} POS_M) + W_S \sum_n (E_{SList} POS_S). \tag{4}$$

ISC: As shown in Algorithm 4, ISC realizes the main functions in the post-warning incentivation stage and monitors AMSC, MFSSC, SCSSC and MSFWSC. *settleReward* is the calculation and implementation function of all rewards and punishments, and its corresponding calculation rules are depicted as formula (5). In order to restrict malicious Vs instantly, their credit scores are updated in real-time in SCSSC. After the early warning finishes, based on the atomicity of smart contracts, *settleReward* will reward all participants at the same time to ensure fairness and security. For cryptocurrency, we simply set up that all rewards come from deposits. However, when all Vs are honest, they may get the rewards equal to their initial deposits, which inevitably leads to the loss of their motivation for validations. And this can be easily solved when governments or other organizers provide additional bonuses for these early warning projects. For credit scores, we set up more detailed incentive mechanisms and prove the effectiveness in Section 4.2.3.

$$\begin{array}{ll}
\textbf{For each} & F_i \in sucFList,\ S_j \in sucSList,\ V_k \in sucVList: \\
& R_{F_i} = \frac{\alpha_{F_i}}{\Sigma_{sucFList}\alpha_{F_i}} F_{dep}, \quad C_{F_i} = C_{F_i} + C_{F+}; \\
& R_{S_j} = \frac{\alpha_{S_j}}{\Sigma_{sucSList}\alpha_{S_j}} S_{dep}, \quad C_{S_j} = C_{S_j} + C_{S+}; \\
& R_{V_k} = R_{V_k} + \frac{1}{len(sucVList)} V_{dep}, \quad C_{V_k} = C_{V_k}; \\
\textbf{For each} & F_i \in puniFList, \quad S_j \in puniSList, \quad V_k \in puniVList: \\
& R_{F_i} = -D_{F_i}, \quad C_{F_i} = C_{F_i} - C_{F-}; \\
& R_{S_j} = -D_{S_j}, \quad C_{S_j} = C_{S_j} - C_{S-}; \\
& R_{V_k} = -D_{V_k}, \quad C_{V_k} = C_{V_k}; \\
\textbf{Last},\ \textbf{for each} & p \in FList \cup SList \cup VList: \\
& FU_{ISC \to p} = D_p + R_p.
\end{array} \tag{5}$$

Here, we discuss the effects of the preset parameters in Table 2. $C_{p+}, C_{p-}, D_{rg}, D_{rq}$ and C_{rw} are parameters determined by incentive mechanisms. The larger the C_{p-}, D_{rg} and D_{rq} are set, the higher the costs of malicious behaviors will be. Especially, when $C_{p-} \gg C_{p+}$, any malicious behavior might lead to the rapid decrease of credit scores until it is too low to exceed the preset threshold of the subsequent operations. For instance, a large C_{rw} will immediately prevent dishonest Vs from reviewing new solutions. Thanks to the decentralized blockchain, theoretically, $F_{lim}, S_{lim}, V_{lim}$ have no maximum limit. However, with their increase, the payload of on-chain communications and off-chain model training will also increase, and more efficient consensus mechanisms and model fusion algorithms should be further considered. For example, in small-scale collaboration surveillance, $\theta_{rw} = V_{lim}$ can be set, which means $solution_{S_j}$ should be verified by all Vs. While in the large-scale collaboration, a smaller θ_{rw} inspired by the existing consensus mechanisms, such as Proof of Stake (POS), Delegated Proof of Stake (DPOS) and Practical Byzantine Fault Tolerance (PBFT), is also acceptable [32]. Additionally, a larger θ_{E_S} helps to limit the lower bound of $Model_{S_j}$'s performance and ensure the advantages of collaborative solution sets. At Last,

$\theta_{POS_S}, \theta_A, W_M$ and W_S are parameters set according to early warning rules. Smaller θ_{POS_S} and θ_A help to improve sensitivity, and the trade-off of W_M and W_S help to improve accuracy and timeliness.

Table 3 is the comparison of the typical research works and our framework. First, our proposed framework is the only decentralized organization that aims to break the resource constraints of centralized and semi-centralized systems. After completing registration, identity authentication and mortgage deposit, capable participants all over the world can freely contribute to our framework by using their preferred data and models. Second, since we only use smart contracts to manage the interaction process, and do not rely on specific data and models, the modification of early warning logics and the integration of multiple data and various models can be easily realized without much cost. This feature gives us a high degree of flexibility which can support both indicator-based early warning and event-based early warning. In addition, with the help of LM, our models and data can be encapsulated as verifiable, traceable and tradable digital assets, thus enhancing their credibility and reusability, and making future early warning easier. In terms of decision-making, our framework automatically screens and fuses distributed decision-making using the preset smart contracts without manual moderation, so it can surpass news aggregators and avoid moderator bias. Third, for information access, unlike the existing systems alternatively designed as public or restricted, our framework has optional accessibility. For example, although the Send function is set to publicly record the transmission logs on blockchain for auditability, the transmitted information is custom encrypted with cryptography technology, and thus its details are only visible to the designated receivers. Further, to break the privacy restrictions of collaborative mining, two different surveillance modes are designed in MFSSC and SCSSC to respectively analyze multi-source data with different privacy sensitivity. These improvements provide some significant information sharing and utilization mechanisms for our EWS. Fourth, compared to the volunteer project Influenzanet, we use ISC to implement dynamic rewards in the form of cryptocurrency and credit scores, so as to encourage effective contributions and unite decentralized organizations.

4. Experiments

In this section, we develop the smart contracts designed in Section 3.3 to implement and examine the proposed collaborative EWS. An early warning scenario for COVID-19 based on open-source chest X-ray data sets is set up to comprehensively verify the functions of smart contracts, and we evaluate and analyze the EWS with multiple quantitative and non-quantitative indicators. Experiments show that the proposed EWS with advantages of auditability and universality can successfully unite the distributed medical and social surveillance forces to reduce the risk of decision bias and provide them with fair incentives and trusted information sharing mechanisms.

4.1. Platform and setup

Based on the development architecture Ganache + Truffle, we create a virtual Ethereum blockchain locally and develop the designed smart contracts by using Solidity programming language to implement the proposed collaborative EWS. Web3.js is used to interact with all smart contracts and verify their performance, Python + Keras is used to implement all training, prediction and fusion algorithms of AI models. And by using the library, Keras.model, models or model weights are saved as files of multiple formats and then stored or shared in IPFS. To reduce the execution costs, we extract the common functions in five smart contracts to form a *Helper* contract for others to call, including querying specific members and calculating contribution coefficients. Therefore, six contracts are deployed, i.e., Helper, AMSC, MFSSC, SCSSC, MSFWSC, and ISC.

The experimental scenario is assumed as follows. To surveil and monitor the emerging COVID-19, three medical institutions with privately-held chest X-ray data sets form a federation to realize medical federated surveillance. Three social monitors and three verifiers form a collaboration organization to realize social collaboration surveillance. In the training stage, social monitors submit AI models for the same trusted open-source chest X-ray data set. For these two surveillance modes, we construct a unified COVID-19 chest X-ray data set containing 203 positive samples and 406 negative samples, and then divide it in two ways as shown in Table 4. To fairly compare the performance of individual and collaboration surveillance models, the *VSet* is the unified validation set sampled from all *F*s, and the Test Set is invisible to all *S*s and *V*s. *S*s and *V*s are set to only train and verify solutions on the Train Set.

Our positive samples are collected from a open-source data set released on GitHub, which consists of chest X-ray and CT images of patients who are positive or suspected of COVID-19 or other viral and bacterial pneumonias (MERS, SARS, and

Table 3
The comparison of the proposed framework and typical researches.

Researches	Organization	Flexibility	Type-based	Decision-making	Information Access	Economic Incentive
CIDARS	Centralized	Low	Indicator	Human-moderated	Restricted	Fixed Salary
GPHIN	Centralized	Fair	Event	Human-moderated	Restricted	Fixed Salary
HealthMap	Centralized	Fair	Event	Automatic	Public	Fixed Salary
Influenzanet	Semi-centralized	Low	Event	News Aggregators	Public	Volunteers
Proposed	Decentralized	High	Hybrid	Automatic	Optional	Dynamic Rewards

Table 4
The data set division at medical end and social end.

COVID-19	Complete Data Set	Medical End				Social End	
		F_1	F_2	F_3	*VSet*	Train Set	Test Set
Positive	203	58	58	87	6 + 6+9 = 21	163	40
Negative	406	116	116	174	12 + 12 + 18 = 42	326	80

ARDS) [7]. We downloaded the data set on July 28, 2020, and screened out all the positive samples of COVID-19 in posteroanterior (PA) view, which has up to 203 samples. Then, to collect the negative samples, we randomly sampled 406 samples tagged as Normal in PA view from an open chest X-ray data set on Kaggle and formed the complete data set [33]. It is worth noting that our data set is only constructed to demonstrate and analyze the proposed EWS, based on which all of the obtained models lack clinical study and can not really diagnose [29,50]. In practical application, the EWS can choose more trustable data sets with smaller bias and more interpretable models with better clinical research.

Additionally, due to the inherently limited computing ability of Ethereum smart contracts, such as only supporting 8–256 signed or unsigned integer variables, failing to compute floating-point numbers, and lacking complete mathematical libraries, we have adopted some development strategies, including calling a verified contracts library named SafeMath for preventing overflow, expanding the parameters, and transforming logarithmic functions into polynomials, For example, $\alpha_{F_i} = getContri(Acc_{F_i}) = \frac{1}{2}\ln\frac{Acc_{F_i}}{1-Acc_{F_i}} \approx \sum_{m=1}^{\infty}\frac{1}{2m-1}(2Acc_{F_i}-1)^{2m-1}$. To avoid fraudulent and other malicious behaviors, all functions in contracts have strict inspection conditions, so they can only be performed when callers obtain the required authentication of identity, time, and calling mode. Finally, in order to save the execution costs, some data is recorded by triggering “Event” instead of creating new parameters.

4.2. Evaluation and discussion

4.2.1. Accuracy

We examine the accuracy of the proposed system from three aspects: Medical End, Social End, and fusion results.

Medical End: Assuming that all *F*s train CNN models with the same structure, F_3 has twice the data of F_1 and F_2. We tested the accuracy of local models, the average federated model, and the weighted federated model on the *VSet* and the complete data set. The results are shown in Fig. 3, and all α_{F_i}s are shown in Table 5. On the complete data set, the performance of F_1, F_2 and F_3 is similar, both federated models are better than local models, and the weighted model has the highest accuracy. On the VSet, the performance of F_1, F_2 and F_3 has a large gap, both federated models are between F_1 and F_3, but the weighted model is better than the average model. This shows that when local models have good and similar performance, federated collaboration is beneficial to improve federated models and thus the warning accuracy. When local models contain both good and bad models with a large gap, federated models will inevitably be affected and deteriorated, but in both cases, the weighted model can better approximate the good models. Besides, when federated models are equally shared among all members, the real contributors may fail to benefit from the cooperation and the lazy members may easily take a free ride, which means that FL without additional incentives is an unfair collaboration, leading to the loss of contribution motivation and the dissolution of the federation. For this issue, ISC sets that all *F*s divide the total deposits F_{dep} in proportion to their α_{F_i}, so as to provide additional economic incentives proportional to the real contribution and maintain the fairness and sustainability of cooperation.

Social End: For the same trusted open-source data set, all *S*s can submit heterogeneous models as their individual surveillance solutions. Assuming that three *S*s submit CNN, LSTM, and BiLSTM models respectively, we discuss two cases that S_1 submits a good CNN model and S_1’ submits a bad CNN model. The accuracy of individual solutions, the average collaborative

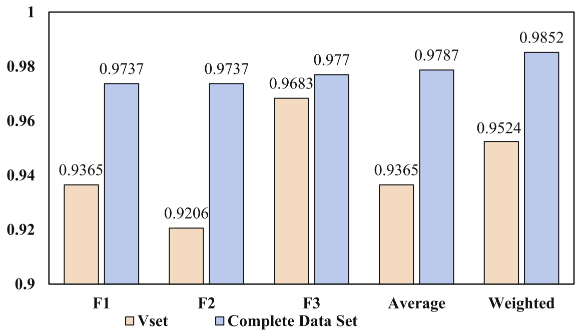


Fig. 3. The accuracy of local models, the average federated model, and the weighted federated model on the *VSet* and the complete data set.

Table 5
The contribution coefficients of federation members.

Participants	F_1	F_2	F_3
α_{F_i}	1.3456	1.2255	1.7089

solution set, and the weighted collaborative solution set on the Train Set and Test Set in two cases are shown in Fig. 4(a) (b), and all α_{S_j}s are shown in Table 6. Since the open-source Train Set is visible to all social participants, we do not divide a special validation set from the Train Set for *V*s, and *S*s may submit overfitting models. In practical application, the early warning project can be initiated by organizers with trusted surveillance data, and they can choose to prepare special validation sets to prevent strategic behaviors. Because this mechanism has been discussed and implemented in our previous work [40], for simplicity, this experiment just assumes that all submitted models are not overfitting. As shown in Fig. 4, when all individual solutions are good, social collaboration can achieve better surveillance performance, regardless of average or weighted set. When individual solutions contain bad models, the collaboration will be affected. However, similar to Medical End, the weighted set can better exclude the negative influence and approximate to good solutions.

Both experiments on Medical End and Social End show that the collaboration can improve the performance of fusion model under certain conditions. This is because as shown in formula (1) and (3), their trainings of fusion surveillance models are essentially completed off the blockchain based on FL and EL, and smart contracts designed to control interaction do not affect models' performance. In other words, these certain conditions that can improve the fusion performance are exactly the effective methods to improve federated models and ensemble models in FL and EL. Therefore, by learning from the recent advances of FL and EL, the advantages of our fusion models and collaborative early warning can be further ensured. In this paper, we simply adopt the above efficient weighted fusion methods to demonstrate and analyze our system.

Fusion results: We test the medical weighted federated model and the social weighted collaborative solution set on the complete data set, and get their misclassified sample sets as shown in Fig. 5, in which the serial numbers of negative samples are less than or equal to 405, and the serial numbers of positive samples are greater than 405. Due to their different data preferences, by adjusting W_M and W_S, the misclassified samples outside their intersection can be identified, so as to improve the accuracy and timeliness of early warning.

For instance, when $m = n = 1$ and the accuracy is taken as the evaluation indicator, the formula to determine whether to alert is as formula (6). When the early warning rule is set to alert whenever a positive sample is detected at an end, the corresponding mathematical expression is as formula (7), and the solutions of W_M, W_S, and θ_A are as formula (8). Then, if we take Acc_{FList} as the accuracy of the weighted federated model on *VSet*, i.e., 0.9524, and take Acc_{SList} as the weighted estimated accuracy as shown in formula (9) because of the invisibility of Test Set, i.e., 0.9782, a feasible solution is $(\theta_A, W_M, W_S) = (0.4, 0.5, 0.5)$. At this time, compared with the early warning rule which is set to alert whenever a positive sample is detected at both ends, the proposed EWS can additionally give correct early warning to the samples in the misclassified set, including $\langle 436 \rangle, \langle 468 \rangle, \langle 480 \rangle, \langle 525 \rangle, \langle 550 \rangle$ and $\langle 553 \rangle$, which improves the performance. When MSFWSC needs to fuse the monitoring results of multiple medical federations and social collaboration organizations, the formula (6) can be easily extended for the combinatorial optimization of weights and thresholds to obtain better accuracy and timeliness of early warning.

$$W_M(Acc_{FList}POS_M) + W_S(Acc_{SList}POS_S) \geqslant \theta_A, \quad \textbf{where} \quad W_M + W_S = 1.$$

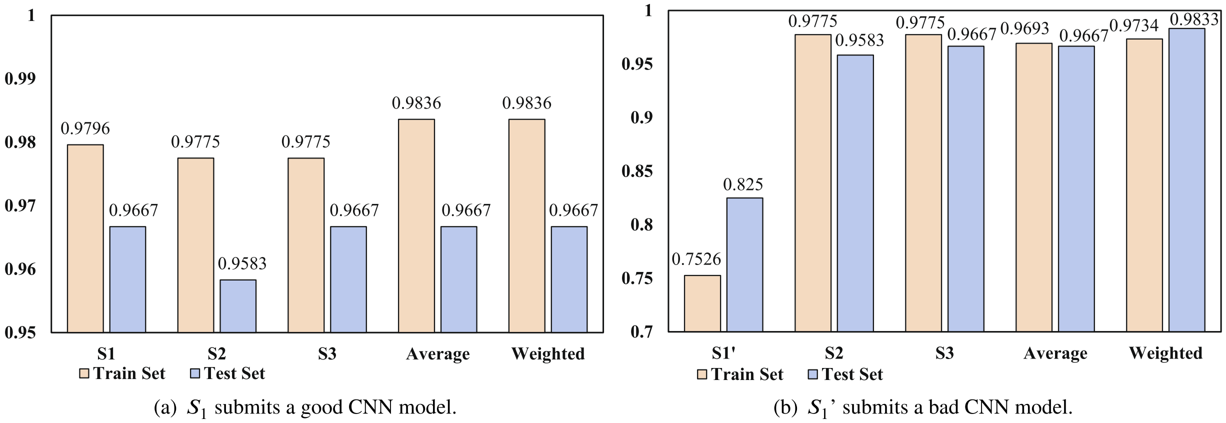
$$\begin{cases} W_M Acc_{FList} + W_S Acc_{SList} \geqslant \theta_A, & \textit{both positive} \\ W_M Acc_{FList} * 0 + W_S Acc_{SList} * 0 < \theta_A, & \textit{both negative} \\ W_M Acc_{FList} \geqslant \theta_A, & \textit{only positive at Medical End} \\ W_S Acc_{SList} \geqslant \theta_A, & \textit{only positive at Social End}. \end{cases} \tag{7}$$

$$\begin{cases} 0 < \theta_A \leqslant \frac{Acc_{FList}Acc_{SList}}{Acc_{FList}+Acc_{SList}} \\ \frac{\theta_A}{Acc_{SList}} \leqslant W_S \leqslant \frac{Acc_{FList}-\theta_A}{Acc_{FList}} \\ W_M = 1 - W_S. \end{cases} \tag{8}$$

$$Acc_{SList} = \sum_{len(sucSList)} \frac{\alpha_{S_j}}{\sum_{len(sucSList)} \alpha_{S_j}} Acc_{S_j}. \tag{9}$$

4.2.2. Collaboration costs

Any execution of programming segments in Ethereum will trigger a payment for computing resources calculated by using the unit called gas according to the preset rules [57]. The actual transaction costs $Cost_{Eth} = gas * \ gasPrice$, where *gasPrice* is the exchange rate between Ethereum's cryptocurrency *Eth* and gas, and can be set arbitrarily by senders. The higher is the



(a) S_1 submits a good CNN model.

(b) S_1' submits a bad CNN model.

Fig. 4. The accuracy of individual solutions, the average collaborative solution set, and the weighted collaborative solution set on the Train Set and Test Set in two cases.

Table 6
The contribution coefficients of social monitors.

Participants	S_1	S_2	S_3	S_1’
α_{S_j}	1.9346	1.8859	1.8859	0.5561

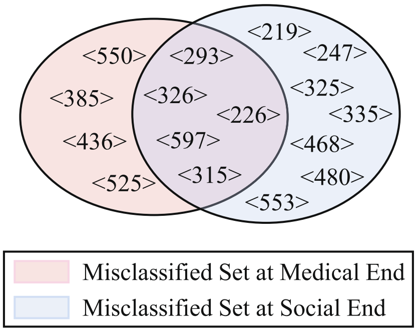


Fig. 5. Misclassified sets of Medical End and Social End on the complete data set.

gasPrice, the faster will the transaction be packaged into the blockchain. The gas consumption of smart contract deployments and executions are shown in Tables 7 and 8, respectively. Considering that the size of parameters, the number of cycles, the order of invoking, and the complexity of calculations will all affect the gas consumption, the data in Tables 7 and 8 are the average values obtained from multiple repeated experiments. Because our smart contracts focus on dealing with the cooperative relationships among all participants and most complex computations are implemented in an off-chain fashion, the operation comsuming the most gas is deploying the six smart contracts, i.e., 12,739,996 gas in sum. Taking $gasPrice = 2 \times 10^{10} wei$ as an example, the consumption of deployment is 0.2548 *Eth*, and according to *Eth*’s historical highest and lowest prices, $1Eth = \$2036.29$ and $1Eth = \$0.4208$ [8], the corresponding costs are \$518.8465 and \$0.1072, respectively. Both of them are far cheaper than the establishment of any centralized EWS that can support the same kind of distributed collaboration. With the future upgrade of the Ethereum platform, the reduction of gas costs, and the adoption of consortium blockchain without execution costs, our costs can be expected to be further reduced.

Another collaboration cost we consider is the computational time. Besides the original model training time, the computational time includes the contract execution time to realize on-chain interactions. In the blockchain network, users send

Table 7
The deployment costs of smart contracts.

Smart contracts	Helper	AMSC	MFSSC	SCSSC	MSFWSC	ISC
Deployment costs (gas)	561988	1931692	2612793	4916691	609052	2107780

Table 8
The execution costs of smart contracts.

AMSC	1. Register	47045	2. Choose Role	67945	3. Delete Role	60841
MFSSC	1. Publish $VSet_{F_i}$	60892	2. Publish $Model_{F_i}$	103294	3. Publish $Model_{FList_{F_i}}$	85960
	3 + 4. Publish Last $Model_{FList_{F_i}}$ & Merge $Model_{FList}$			220703	5. Monitor & Report RES_M	117588
MFSSC	1. Share Resources	83093	2 1). Publish $Model_{S_j}$	150063	2 2). Send k_{S_j}	65173
	3. Review $Model_{S_j}$	125860	3 + 4 1) 2) 3). S_j's Last Review & Merge E_{S_j}			338450
	3 + 4. Last S_j's Last Review & Merge Last E_{S_j} & Estimate E_{SList}			453995	5. Monitor & Report RES_S	94798
MSFWSC	1. Fuse M&S Monitoring Result RES_{MS}					74541
ISC	2. Reward Medical Federations		377964	2. Reward Social Collaboration Organizations		934826

new transactions to invoke smart contracts, miners sequentially verify, execute and package these transactions to prepare a new block, and the contract execution will only take effect after the new block is added to the main chain. In other words, transactions sent asynchronously by distributed participants but packaged into the same block can be regarded as having the same computational time, namely, block time. Therefore, taking federated surveillance in MFSSC as an example, assuming that in all steps of Algorithm 2, the time differences among all Fs invoking MFSSC for the same phased operations are less than one block time, and their transactions are immediately packaged into the same added block without pending, then, the additional computational time of a model iteration (Step 2 to Step 4) is two block times because Step 3 automatically goes to Step 4 when all Fs finish. Similarly, that in SCSSC (Step 2 to Step 4 in Algorithm 3) is also two block times. Further, since Fs and Ss only simultaneously monitor emerging cases based on the trained models in the future long-term monitoring, the additional contract execution time for each later collaborative early warning is actually two block times (Step 5 in Algorithm 2/3 + Step 1 in Algorithm 4). Considering that the average block time in the virtual blockchain can be easily customized, we refer to that of the existing Ethereum, which is about 13 s, and the two block times are about 26 s, so it is still attractive to most existing EWSs [9].

4.2.3. Incentive mechanisms

For credit scores, our incentive mechanisms are designed as follows:

1) When social participants upload a piece of data and model, they can get 20 scores. When S publishes a $solution_{S_j}$, he/she should upload corresponding data and models at the same time, and then gets 20 scores in total;
2) V with credit scores greater than 10 can participate in the validation, and he/she needs to consume 10 scores to review a solution and its corresponding data and models, i.e., $C_{rw} = 10$;
3) If a $solution_{S_j}$ is determined to be invalid after the consensus of all its verifiers, S_j will be punished with 30 scores, and otherwise be rewarded with 10 scores, i.e., $C_{S-} = 30, C_{S+} = 10, C_{S-} > C_{S+}$;
4) Taking Vs' consensus as the correct validation results, every V can get 20 scores for one correct validation, and lose 30 scores for one false validation, i.e., $C_{V+} = 20, C_{V-} = 30, C_{V-} > C_{V+}$.

In this section, we carry out a numerical simulation experiment to verify the effectiveness of the above incentive mechanisms. Suppose that there are six different participants at the Social End who serve as both social monitors and verifiers. Every participant has 500 solutions, and each solution corresponds to one piece of data and model. In each iteration, they publish a random number of solutions until all their solutions are published. At the same time, they verify a random number of solutions published by other participants, but can only verify a specific solution once. Every participant's initial credit score is 0. We divide the participants into three types: honest, malicious and lazy (or random), and their settings are as follows:

In each iteration:

- Honest participants have a probability of 0.9 to publish valid solutions and give correct verification;
- Lazy participants have a probability of 0.5 to publish valid solutions and give correct verification;
- Malicious participants have a probability of 0.1 to publish valid solutions and give correct verification.

We set up two experimental scenarios. In both scenarios, 1, 2, 3, 4 are honest participants, and 5 is a lazy participant (baseline). 1) 6 is a malicious participant. As shown in Fig. 6 (a), the credit scores of 1, 2, 3 and 4 gradually increase until all solutions are published. The credit score of 5 fluctuates around 0 and is much lower than honest participants. The credit score of 6 is negative, indicating that he/she has been punished. This shows that the above incentive mechanisms can effectively encourage honest contribution, and reduce motivations for malicious behaviors. 2) 6 is honest in the first three iterations and becomes malicious from the fourth iteration. As shown in Fig. 6 (b), the credit score of 6 increases in the first three iterations, declines sharply in the fourth iteration, and finally stabilizes to negative much lower than the lazy participant.

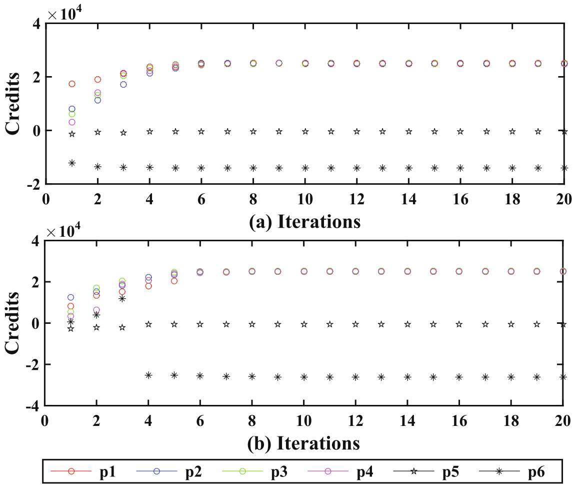


Fig. 6. The numerical analysis of incentive mechanisms.

This shows that the above incentive mechanisms can help detect malicious behaviors and punish malicious participants in time. Hence, the proposed EWS can obtain certain anti-attack ability by restricting operation authorities based on credit scores. These experimental results are consistent with our discussion of C_{p+}, C_{p-} and C_{rw} in Section 3.3.

Considering that infectious disease surveillance and early warning usually requires the long-term collaboration of a large number of geographically distributed contributors with different expertise knowledge, EWS with built-in incentive mechanisms based on blockchain, smart contracts and token economics are conducive to fairly quantifying the contribution, accurately evaluating the shared information, and continuously promoting the collaboration of global capable individuals, institutions and organizations.

4.2.4. Traceability and auditability

In our proposed framework, all participants realize collaborative early warning by invoking smart contracts, and their interactive records will be publicly stored on the blockchain, which enables its traceability and auditability. The process of forward fusion and reverse tracing for early warning information is illustrated in Fig. 7. The black arrow indicates the forward fusion of distributed early warning information described in Section 3.2. Whenever the final fusion results exceed the preset thresholds, the EWS sends out an early warning signal. Then, according to the red arrow, the monitoring results leading to the early warning and their corresponding hospitals, participants, surveillance data, and models can be traced back, and thus the associated suspected cases can be located. Finally, epidemiologists can conduct strict epidemiological investigations or clinical diagnosis on these cases to formally determine the confirmed cases and issue official early warning signals. This traceability and auditability can not only help to quickly locate suspected cases and enrich decision-making information, but also help detect and exclude unreliable cooperators in advance.

4.2.5. Trusted information sharing and decentralized decision-making

Considering the privacy, unreliability, and fragmentation of multi-source surveillance resources, it is difficult for traditional EWSs to share significant information, solidify decision-making basis, and incorporate untrusted participants. Therefore, in the sharing market at the Social End, the proposed EWS certifies, verifies, and traces all participants, data, models, and solutions in the long-term collaboration, so as to enhance the reliability of shared resources and the credibility of distributed decision-making. Also, smart contracts with preset rules serve as decentralized software-defined agents to monitor, verify, screen out, and fuse multi-party decisions in a real-time fashion and automatically alert, thereby avoiding the deci-

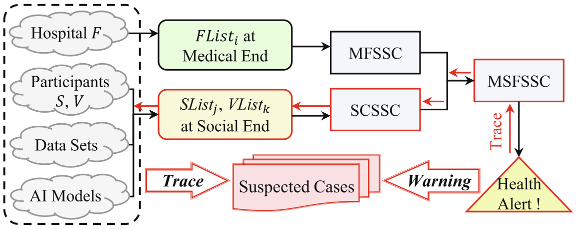


Fig. 7. The forward fusion and reverse tracing process of early warning information.

sion bias. For private data such as medical records, the proposed EWS adopts a privacy computing architecture, i.e., blockchain-based FL, to improve the performance with privacy protection. And for significant information sharing, it adopts cryptography technology and authority management to encrypt transmission and limit permissions, and thus can achieve private cooperation.

4.2.6. Flexibility and universality

Although we take an early warning scenario for COVID-19 based on open-source chest X-ray data sets as an example to implement our proposed EWS, the system is flexible and universal. By replacing the surveillance data and models, it can monitor various infectious diseases from multiple aspects. For instance, for the known diseases, their symptom surveillance models' training sets can adopt CT images, electronic medical records, and infectious disease notifications according to their characteristics. While for unknown diseases, semi-supervised or unsupervised models such as clustering [30], principal component analysis (PCA) [55] and generative adversarial networks (GAN) [43,53] can be adopted for anomaly detection. Additionally, by modifying the fusion algorithms, early warning rules, and incentive mechanisms in the smart contracts, the system can support more collaboration modes, and the design of smart contracts is applicable to different types of blockchain.

5. Future work

In the future, our work can be extended from the following three aspects:

First, expanding the system to monitor more infectious diseases, support more collaboration modes, and attract more distributed contributors. In this paper, we have demonstrated that smart contracts can encapsulate and execute rules and mechanisms. Therefore, authoritative organizations such as the CDC can similarly define the formal early warning process with different collaborative modes and incentive mechanisms by developing and issuing a variety of formally verified standardized smart contracts, so as to give full play to the crowd intelligence of the society. In addition, the traceable credit and quality records in LM can help participants with limited capability choose honest collaborators and trade high-quality resources, thereby breaking the barriers of surveillance and early warning. And based on these credit and quality records, the fusion weights can also be calculated quantitatively and updated in real-time.

Second, improving security and privacy based on homomorphic encryption [1], trusted computing, and future smart contracts with better performance. Due to the limited capabilities of the current smart contracts, we introduce additional verifiers for off-chain validation at Social End. Whenever verifiers can obtain complete data and models, there is a possibility of leaking and stealing private information. Although effective incentive mechanisms can greatly reduce such motivation, the on-chain model validation and fusion in smart contracts based on homomorphic encryption and trusted computing is still an attractive solution. Homomorphic encryption can maintain parameter privacy when fusing weights, but it requires a series of conversions before being used in complex models such as neural networks. Mendis et al. has made a preliminary attempt. The results show that this method needs complex computation and depends on specific models and homomorphic encryption algorithms, which inevitably lead to the loss of universality [31]. Similar to the idea of Ethereum Layer 2 initiative [48], specialized machine clusters can be built based on the trusted execution environment (TEE) [44] to encapsulate standard algorithms for validation and fusion, complete all the calculations off chain, and only upload the results back to smart contracts, thereby reducing the on-chain computing payload.

Third, enhancing the public's prevention of unknown diseases based on recommendation algorithms [4] and search engine systems [13]. With the development of unsupervised learning algorithms, the proposed EWS is expected to detect abnormal phenomena in the early stage of unknown diseases, such as the sharp increase of abnormal chest X-ray images at Medical End before the outbreak of COVID-19, which is an important indicator of both COVID-19 and SARS. Although the system can not make a clinical diagnosis, it can be further combined with recommendation and search engine algorithms to determine the disease category according to the characteristics of indicator data, and finally recommend the corresponding universal health suggestions. For example, for the unknown respiratory diseases characterized by a pulmonary infection such as COVID-19 and SARS, the system can automatically issue early warning signals to the public as well as some practical prevention advice for respiratory diseases, such as wearing masks, washing hands and keeping social distance, and thus realize early prevention of unknown diseases.

6. Conclusion

Aiming at reducing the risk of decision-making errors and improving the performance of early warning, this article proposes a novel framework of collaborative early warning for COVID-19 based on blockchain and smart contracts. This framework features blockchain-based secured and private collaboration environments, scalable warning and incentive mechanisms executed in smart contracts, as well as a variety of encapasulated AI models for early warning. In order to fully combine distributed surveillance forces and integrate multi-source surveillance resources, our proposed system supports two different surveillance modes and fuses their monitoring results on emerging cases to alert, which includes medical federation surveillance based on FL and social collaboration surveillance based on LM. We preliminarily implement and analyze our proposed system based on Ethereum and IPFS. The results show that it has the advantages of decentralized decision-making, fairness, auditability and universality, and also has the potential to further help the early warning and prevention of unknown infectious diseases.

CRediT authorship contribution statement

Liwei Ouyang: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing. **Yong Yuan:** Conceptualization, Writing - original draft, Writing - review & editing, Supervision. **Yumeng Cao:** Writing - original draft, Writing - review & editing. **Fei-Yue Wang:** Conceptualization, Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported in part by the National Key R&D Program of China (2018AAA0101401), the Science and Technology Development Fund, Macau SAR (File No. 0050/2020/A1), and the National Natural Science Foundation of China (61533019, 71702182).

References

[1] A. Acar, H. Aksu, A.S. Uluagac, M. Conti, A survey on homomorphic encryption schemes: theory and implementation, ACM Comput. Surveys (CSUR) 51 (2018) 1–35.

[2] A. Azaria, A. Ekblaw, T. Vieira, A. Lippman, Medrec: using blockchain for medical data access and permission management, in: 2016 2nd International Conference on Open and Big Data (OBD), IEEE, 2016, pp. 25–30..

[3] J. Benet, Ipfs-content addressed, versioned, p2p file system, 2014. arXiv preprint arXiv:1407.3561..

[4] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132.

[5] S. Cao, G. Zhang, P. Liu, X. Zhang, F. Neri, Cloud-assisted secure ehealth systems for tamper-proofing ehr via blockchain, Inf. Sci. 485 (2019) 427–440.

[6] J. Choi, Y. Cho, E. Shim, H. Woo, Web-based infectious disease surveillance systems and public health perspectives: a systematic review, BMC Public Health 16 (2016) 1–10.

[7] J.P. Cohen, P. Morrison, L. Dao, Covid-19 image data collection, 2020. arXiv 2003.11597 URL:https://github.com/ieee8023/covid-chestxray-dataset..

[8] CoinMarketCap, Historical data for ethereum. [EB/OL]. URL:https://coinmarketcap.com/currencies/ethereum/historical-data/ Accessed February 28, 2021..

[9] etherchain, The ethereum block chain explorer. [EB/OL]. URL:https://etherchain.org/ Accessed February 28, 2021..

[10] C.C. Freifeld, K.D. Mandl, B.Y. Reis, J.S. Brownstein, Healthmap: global infectious disease monitoring through automated classification and visualization of internet media reports, J. Am. Med. Inf. Assoc. 15 (2008) 150–157.

[11] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1997) 119–139.

[12] J. Ginsberg, M.H. Mohebbi, R.S. Patel, L. Brammer, M.S. Smolinski, L. Brilliant, Detecting influenza epidemics using search engine query data, Nature 457 (2009) 1012–1014.

[13] D.A. Grossman, O. Frieder, Information Retrieval: Algorithms and Heuristics, vol. 15, Springer Science & Business Media, 2012.

[14] HealthMap, The home page of healthmap. [EB/OL]. URL:https://healthmap.org/zh/ Accessed February 24, 2021..

[15] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.

[16] C. Hong, J. Yu, D. Tao, M. Wang, Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval, IEEE Trans. Industr. Electron. 62 (2014) 3742–3751.

[17] C. Hong, J. Yu, J. Zhang, X. Jin, K.H. Lee, Multimodal face-pose estimation with multitask manifold deep learning, IEEE Trans. Industr. Inf. 15 (2018) 3952–3961.

[18] A. Hulth, G. Rydevik, Get well: an automated surveillance system for gaining new epidemiological knowledge, BMC Public Health 11 (2011) 1–8.

[19] Influenzanet, The home page of influenzanet. [EB/OL]. URL:http://influenzanet.info Accessed February 24, 2021..

[20] H. Kim, J. Park, M. Bennis, S.L. Kim, Blockchained on-device federated learning, IEEE Commun. Lett. (2019).

[21] J. Konečný, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, 2016. arXiv preprint arXiv:1610.05492..

[22] T.T. Kuo, H.E. Kim, L. Ohno-Machado, Blockchain distributed ledger technologies for biomedical and health care applications, J. Am. Med. Inform. Assoc. 24 (2017) 1211–1220.

[23] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (1998) 2278–2324.

[24] J. Lederberg, M.A. Hamburg, M.S. Smolinski, et al, Microbial Threats to Health: Emergence, Detection, and Response, National Academies Press, 2003.

[25] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (2020) 1234–1240.

[26] C. Lin, D. He, X. Huang, X. Xie, K.K.R. Choo, Blockchain-based system for secure outsourcing of bilinear pairings, Inf. Sci. 527 (2020) 590–601.

[27] J.P. Linge, R. Steinberger, T. Weber, R. Yangarber, E. van der Goot, D. Al Khudhairy, N. Stilianakis, Internet surveillance systems for early alerting of health threats, Eurosurveillance 14 (2009) 19162.

[28] M.O. Lwin, S. Vijaykumar, O.N.N. Fernando, S.A. Cheong, V.S. Rathnayake, G. Lim, Y.L. Theng, S. Chaudhuri, S. Foo, A 21st century approach to tackling dengue: Crowdsourced surveillance, predictive mapping and tailored communication, Acta Trop. 130 (2014) 100–107.

[29] G. Maguolo, L. Nanni, A critic evaluation of methods for covid-19 automatic detection from X-ray images, 2020. arXiv:2004.12823..

[30] G. McLachlan, Cluster analysis and related techniques in medical research, Stat. Methods Med. Res. 1 (1992) 27–48.

[31] G.J. Mendis, Y. Wu, J. Wei, M. Sabounchi, R. Roche, Blockchain as a service: a decentralized and secure computing paradigm, 2018. arXiv preprint arXiv:1807.02515..

[32] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, C. Qijun, A review on consensus algorithm of blockchain, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2017, pp. 2567–2572..

[33] P. Mooney, Chest X-ray images (pneumonia). [EB/OL]. URL:https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia Accessed November 25, 2020..

[34] S.S. Morse, Global infectious disease surveillance and health intelligence, Health Aff. 26 (2007) 1069–1077.

[35] E. Mykhalovskiy, L. Weir, The global public health intelligence network and early warning outbreak detection, Can. J. Public Health 97 (2006) 42–44.

[36] S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system. [EB/OL]. URL:https://bitcoin.org/bitcoin.pdf Accessed November 25, 2020..
[37] S.P. van Noort, C.T. Codeco, C.E. Koppeschaar, M. Van Ranst, D. Paolotti, M.G.M. Gomes, Ten-year performance of influenzanet: Ili time series, risks, vaccine effects, and care-seeking behaviour, Epidemics 13 (2015) 28–36.
[38] M. Odlum, S. Yoon, What can we learn about the ebola outbreak from tweets?, Am J. Infect. Control 43 (2015) 563–571.
[39] L. Ouyang, Y. Yuan, F.Y. Wang, A blockchain-based framework for collaborative production in distributed and social manufacturing, in: 2019 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), IEEE, 2019, pp. 76–81.
[40] L. Ouyang, Y. Yuan, F.Y. Wang, Learning markets: an ai collaboration framework based on blockchain and smart contracts, IEEE Internet Things J. (2020).
[41] C. Paquet, D. Coulombier, R. Kaiser, M. Ciotti, Epidemic intelligence: a new framework for strengthening disease surveillance in europe, Eurosurveillance 11 (2006) 5–6.
[42] ProMed-mail, The home page of promed-mail. [EB/OL]. URL:https://promedmail.org/ Accessed February 24, 2021..
[43] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 2016. arXiv:1511.06434..
[44] M. Sabt, M. Achemlal, A. Bouabdallah, Trusted execution environment: what it is, and what it is not, in: 2015 IEEE Trustcom/BigDataSE/ISPA, IEEE, 2015, pp. 57–64.
[45] M. Salathé, Digital pharmacovigilance and disease surveillance: combining traditional and big-data systems for better public health, J. Infect. Dis. 214 (2016) S399–S403.
[46] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (1997) 2673–2681.
[47] M.S. Smolinski, A.W. Crawley, K. Baltrusaitis, R. Chunara, J.M. Olsen, O. Wójcik, M. Santillana, A. Nguyen, J.S. Brownstein, Flu near you: crowdsourced symptom reporting spanning 2 influenza seasons, Am. J. Public Health 105 (2015) 2124–2130.
[48] J. Stark, Making sense of ethereum's layer 2 scaling solutions: State channels, plasma, and truebit. [EB/OL]. URL:https://medium.com/l4-media/making-sense-of-ethereums-layer-2-scaling-solutions-state-channels-plasma-and-truebit-22cb40dcc2f4 Accessed November 25, 2020..
[49] N. Szabo, Smart contracts. [EB/OL]. URL:https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html Accessed November 4, 2020..
[50] E. Tartaglione, C.A. Barbano, C. Berzovini, M. Calandri, M. Grangetto, Unveiling covid-19 from chest X-ray with deep learning: a hurdles race with small data, Int. J. Environ. Res. Public Health 17 (2020) 6933, https://doi.org/10.3390/ijerph17186933, URL:https://doi.org/10.3390/ijerph17186933.
[51] S. Varela-Santos, P. Melin, A new approach for classifying coronavirus covid-19 based on its manifestation on chest X-rays using texture features and neural networks, Inf. Sci. 545 (2020) 403–414.
[52] F.Y. Wang, Y. Yuan, J. Zhang, R. Qin, M.H. Smith, Blockchainized internet of minds: a new opportunity for cyber-physical-social systems, IEEE Trans. Comput. Social Syst. 5 (2018) 897–906.
[53] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, F.Y. Wang, Generative adversarial networks: introduction and outlook, IEEE/CAA J. Autom. Sin. 4 (2017) 588–598.
[54] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, F.Y. Wang, Blockchain-enabled smart contracts: architecture, applications, and future trends, IEEE Trans. Syst., Man Cybern.: Syst. 49 (2019) 2266–2277.
[55] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometr. Intell. Lab. Syst. 2 (1987) 37–52.
[56] Z.S. Wong, J. Zhou, Q. Zhang, Artificial intelligence for infectious disease big data analytics, Infect., Disease Health 24 (2019) 44–48.
[57] G. Wood et al, Ethereum: aa secure decentralised generalised transaction ledger, Ethereum Project Yellow Paper 151 (2014) 1–32.
[58] WorldHealthOrganization, Who coronavirus disease (covid-19) dashboard. [EB/OL]. URL:https://covid19.who.int/ Accessed November 25, 2020..
[59] Q. Xia, E.B. Sifah, A. Smahi, S. Amofa, X. Zhang, Bbds: blockchain-based data sharing for electronic medical records in cloud environments, Information 8 (2017) 44.
[60] W. Yang, Early Warning for Infectious Disease Outbreak: Theory and Practice, Academic Press, 2017.
[61] W. Yang, Z. Li, Y. Lan, J. Ma, L. Jin, S. Lai, Y. Liao, W. Lv, Q. Sun, J. Wang, China infectious diseases automated-alert and response system (cidars), in: Early Warning for Infectious Disease Outbreak, 2017, Elsevier, pp. 133–161..
[62] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing, IEEE Comput. Intell. Mag. 13 (2018) 55–75.
[63] J. Yu, Y. Rui, B. Chen, Exploiting click constraints and multi-view features for image re-ranking, IEEE Trans. Multimedia 16 (2013) 159–168.
[64] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, IEEE Trans. Image Process. 23 (2014) 2019–2032.
[65] J. Yu, D. Tao, M. Wang, Y. Rui, Learning to rank using user clicks and visual features for image retrieval, IEEE Trans. Cybern. 45 (2014) 767–779.
[66] V.L. Yu, L.C. Madoff, Promed-mail: an early warning system for emerging diseases, Clin. Infect. Diseases 39 (2004) 227–232.
[67] Y. Yuan, X. Ni, S. Zeng, F. Wang, Blockchain consensus algorithms: the state of the art and future trends, Acta Autom. Sin. 44 (2018) 2011–2022.
[68] Y. Yuan, F.Y. Wang, Blockchain and cryptocurrencies: model, techniques, and applications, IEEE Trans. Syst., Man, Cybern.: Syst. 48 (2018) 1421–1428.
[69] P. Zhang, D.C. Schmidt, J. White, G. Lenz, Blockchain technology use cases in healthcare, in: Advances in Computers, Elsevier, vol. 111, 2018, pp. 1–41..
[70] Y. Zhang, R.H. Deng, X. Liu, D. Zheng, Blockchain based efficient and robust fair payment for outsourcing services in cloud computing, Inf. Sci. 462 (2018) 262–277.

Liwei Ouyang received the B.S. degree in automation from Xi'an Jiaotong University, Xi'an, China, in 2018. She is currently pursuing the Ph.D. degree in social computing with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

Her current research interests include social computing, blockchain and smart contracts.

Yong Yuan received the B.S., M.S., and Ph.D. degrees in computer software and theory from the Shandong University of Science and Technology, Shandong, China, in 2001, 2004, and 2008, respectively.

He is currently a Professor with the School of Mathematics, Renmin University of China. He is also with the Engineering Research Center of Finance Computation and Digital Engineering, Ministry of Education. He has authored over 120 papers published in academic journals and conferences. His current research interests include blockchain, cryptocurrency, and smart contracts.

Dr. Yuan is currently an Associate Editor of the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS and Acta Automatica Sinica. He is the Chair of the IEEE Council on RFID Technical Committee on Blockchain, the Co-Chair of the IEEE SMC Technical Committee on Blockchain, and the Director of the Chinese Association of Automation Technical Committee of Blockchain. He is the Secretary General of the IEEE SMC Technical Committee on Social Computing and Social Intelligence, the Vice Chair of the IFAC Technical Committee on Economic, Business and Financial Systems (TC 9.1), and the Chair of the ACM Beijing Chapter on Social and Economic Computing. He is also the Secretary General of the Chinese Association of Artificial Intelligence Technical Committee on Social Computing and Social Intelligence, and the Vice Director and the Secretary General of the Chinese Academy of Management Technical Committee on Parallel Management.

Yumeng Cao received the B.A. and MTI degrees from Harbin Engineering University, Harbin, China, in 2015 and 2017, respectively. She is currently an engineer with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

Her current research interest includes social computing.

Fei-Yue Wang (S'87-M'89-SM'94-F'03) received his Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined The University of Arizona in 1990 and became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Chinese Talents Program from the State Planning Council, and in 2002, was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory for Management and Control of Complex Systems.

His current research focuses on methods and applications for parallel intelligence, social computing, and knowledge automation. He is a fellow of INCOSE, IFAC, ASME, and AAAS. In 2007, he received the National Prize in Natural Sciences of China and became an Outstanding Scientist of ACM for his work in intelligent control and social computing. He received the IEEE ITS Outstanding Application and Research Awards in 2009 and 2011, respectively. In 2014, he received the IEEE SMC Society Norbert Wiener Award. Since 1997, he has been serving as the General or Program Chair of over 30 IEEE, INFORMS, IFAC, ACM, and ASME conferences. He was the President of the IEEE ITS Society from 2005 to 2007, the Chinese Association for Science and Technology, USA, in 2005, the American Zhu Kezhen Education Foundation from 2007 to 2008, the Vice President of the ACM China Council from 2010 to 2011, the Vice President and the Secretary General of the Chinese Association of Automation from 2008–2018. He was the Founding Editor-in-Chief (EiC) of the International Journal of Intelligent Control and Systems from 1995 to 2000, the IEEE ITS Magazine from 2006 to 2007, the IEEE/CAA JOURNAL OF AUTOMATICA SINICA from 2014–2017, and the China's Journal of Command and Control from 2015–2020. He was the EiC of the IEEE Intelligent Systems from 2009 to 2012, the IEEE TRANSACTIONS ON Intelligent Transportation Systems from 2009 to 2016, and is the EiC of the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS since 2017, and the Founding EiC of China's Journal of Intelligent Science and Technology since 2019. Currently, he is the President of CAA's Supervision Council, IEEE Council on RFID, and Vice President of IEEE Systems, Man, and Cybernetics Society.