Journal of King Saud University – Computer and Information Sciences 35 (2023) 101747

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Secure-fault-tolerant efficient industrial internet of healthcare things framework based on digital twin federated fog-cloud networks

Abdullah Lakhan [a,f,g], Ali Azawii Abdul Lateef [b,i], Mohd Khanapi Abd Ghani [c], Karrar Hameed Abdulkareem [d], Mazin Abed Mohammed [e,f,g,*], Jan Nedoma [f], Radek Martinek [g], Begoña Garcia-Zapirain [h]

[a] Department of Computer Science and Cybersecurity, Dawood University of Engineering and Technology, Karachi, Pakistan
[b] Human Resources Department, University Headquarter, University of Anbar, Ramadi 31001, Anbar, Iraq
[c] Biomedical Computing and Engineering Technologies (BIOCORE) Applied Research Group, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Durian Tunggal 76100, Malaysia
[d] College of Agriculture, Al-Muthanna University, Samawah 66001, Iraq
[e] College of Computer Science and Information Technology, University of Anbar, Ramadi 31001, Anbar, Iraq
[f] Department of Telecommunications, VSB-Technical University of Ostrava, Ostrava, Czech Republic
[g] Department of Cybernetics and Biomedical Engineering, VSB-Technical University of Ostrava, Ostrava, Czech Republic
[h] eVIDA Lab, University of Deusto, 48007 Bilbao, Spain
[i] Deprament of administrative and financial affairs, University Headquarter, University of Anbar, Ramadi, 31001, Anbar, Iraq

ARTICLE INFO

Article history:
Received 29 March 2023
Revised 4 September 2023
Accepted 6 September 2023
Available online 14 September 2023

Keywords:
IoHT
Fault-tolerant
Digital twin
Industry 5.0
Blockchain
SFTS
Fog-cloud networks
CNN

ABSTRACT

The Industrial Internet of Healthcare Things (IIoHT) is the emerging paradigm in digital healthcare. Context-aware healthcare sensors, local intelligent watches, healthcare devices, wireless communication technologies, fog, and cloud computing are all parts of the IIoHT used in healthcare. The ubiquitous healthcare services it provides to its users in practice. However, the current IIoHT healthcare frameworks have security and failure issues in mobile fog and cloud networks where they are spread out. This paper presents the secure, fault-tolerant IIoHT Framework based on digital twin (DT) federated learning-enabled fog-cloud models. The DT is an effective technology that makes virtual copies of servers at different locations. DT integrated with federated learning inside the fog and cloud environments, where the failure of tasks and execution improved for healthcare sensor data. The study aims to reduce processing time and the risk of task failure. The study presents the Secure and Fault-Tolerant Strategies (SFTS)-enabled IIoHT framework that optimizes wearable sensor data and executes it with the minimum offloading and processing delays. Simulation results show that the proposed work minimized the security risk by 40%, failure risk of tasks risk by 50%, and the training and testing time by 39% for sensor data during the execution of mobile fog cloud networks.

1. Introduction

The industrial Internet of Things (IIoT) is a revolutionary paradigm to improve digital productivity in different businesses (e.g., healthcare, manufacturing, and transport) (Lv, 2023). It started with Industry 1.0, which optimized the steam of engine machines. Industry 2.0 has a lot of optimization in distributed electricity in smart cities (Rashid et al., 2022). Industry 3.0 determined the optimal information about ages with depth transformation into results. Industry 4.0 is about automation and intelligence; many static and dynamic systems have been transformed into automation. Initiated by the European Union in 2021, the revolutionary paradigm is known as Industrial 5.0. The objective is to connect wearable and industrial devices with distributed artificial intelligence-based services (Leng et al., 2022). Cloud computing is a remote ser-

* Corresponding author at: College of Computer Science and Information Technology, University of Anbar, Ramadi 31001, Anbar, Iraq.
E-mail addresses: abdullah.lakhan@duet.edu.pk (A. Lakhan), aliazawii@uoanbar.edu.iq (A.A. Abdul Lateef), khanapi@utem.edu.my (M.K. Abd Ghani), Khak9784@mu.edu.iq (K.H. Abdulkareem), mazinalshujeary@uoanbar.edu.iq (M.A. Mohammed), jan.nedoma@vsb.cz (J. Nedoma), radek.martinek@vsb.cz (R. Martinek), mbgarcia-zapi@deusto.es (B. Garcia-Zapirain).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.101747
1319-1578/

vice provider corporation where different providers offer services to the 5.0 industry-enabled industries for execution (Leong et al., 2021).

Many technological developments in healthcare sectors based on 5.0 with cloud computing have recently been seen and implemented in different clinics. Cloud computing offers virtual services to the 5.0 industrial Internet of Healthcare Things (IIoHT), the version of IIoT where remote healthcare monitoring improves the quality of life (Chi et al., 2022). Modern Artificial Intelligence (AI) and machine learning algorithms have a lot of impact in Industry 5.0 to make the healthcare application (e.g., IoT Covid-19) with full automation through different training and testing phases (Konigsburg, 2022; Salman and Geman, 2023). IIoHT is a collection of healthcare sensors equipped with human bodies that monitor rates in real-time. Cloud computing outsources healthcare services at different layers. For instance, fog computing is the cloud paradigm that brings cloud services to the healthcare radio network layer with minimum end-to-end latency (Khoso et al., 2021). The real-time monitoring and huge amount of data generated by sensors with increased users lead to challenges in the IIoHT system for processing. Furthermore, many challenges exist in industrial 5.0-aware IIoHT for healthcare industries, such as resource constraints, failure of nodes, and security, along with the quality of service requirements of applications (Younan et al., 2020).

The digital twin (DT) is an emerging technology offering different physical server replicas through virtualization (Khan et al., 2022; Elayan et al., 2021; Ghita et al., 2020). DT technology has made many contributions to industry-5.0-enabled healthcare technologies. For instance, DT offers components, infrastructure, and resource replicas to the same and different nodes. The goal is to handle vast healthcare sensor requests on other healthcare servers (Volkov et al., 2021; Haleem et al., 2023). The DT reduces the resource constraints issue of healthcare sensors and servers with the virtual replica at various locations. However, security is a critical issue in DT technologies. Blockchain is decentralized, where all autonomous physical and virtual entities can transfer data with validity and transparency (Azzaoui et al., 2021; Jimenez et al., 2020; Akash and Ferdous, 2022; el Azzaoui et al., 2020; Zhang et al., 2020). Blockchain schemes can validate data transactions among physical and virtual servers during their execution without showing abstraction to the users. However, blockchain technology requires a considerable amount of resources for transactions. Therefore, transaction and node failures are common in blockchain-based DT for healthcare applications. The fault-tolerant, efficient DT technologies presented by these studies (Nguyen et al., 2022; Darvishi et al., 2021; Alshathri et al., 2023). However, these techniques did not consider security aspects. Therefore, the fog cloud network has no digital twin-enabled IIoHT system.(i) Due to a distributed system, the failure of resources leads to the failure of healthcare tasks. It is a necessary process. Therefore, fault-tolerant resources must be part of the IoT healthcare system to process the sensory data without failure. (ii) Many things can go awry with resource failure nodes, such as missing deadlines, critical tasks not doing their jobs, and not getting the best results needed. (iii) The sensory data is the challenging component of the IoT-enabled healthcare system, where processing centralized server nodes on big sensory data takes much time for execution. (iv) Resource-efficient security is most important when fog nodes have limited resources at the radio network for processing with minimum end-to-end delays.

In this paper, we are considering the following research questions: (i) Due to a distributed system, resource failure in hospital nodes leads to the complete failure of critical healthcare tasks. Existing DT failure strategies (Nguyen et al., 2022; Darvishi et al., 2021; Alshathri et al., 2023) only focused on replica failure. However, it is not beneficial for large systems like distributed healthcare with many sensors. (ii) Existing DT-enabled healthcare systems did not combine security and fault tolerance. Therefore, there must be a balance between security and failure of tasks and nodes in the DT-enabled healthcare system.

This paper presents the secure, fault-tolerant Industrial Internet of Healthcare Things (IIoHT) system based on digital twin federated fog-cloud models. The study aims to reduce the time needed to process healthcare sensor data for security, task execution, and fault tolerance while using less resources. The paper has the following contributions to the research questions, including federated learning and digital twin technology.

- The study integrated the fog and cloud nodes based on digital twin technology, where all local nodes at different laboratories have replica copies of trained data and processing capability in the same runtime environment.
- We integrated the different kinds of healthcare sensors in the human body. We connected them with mobile devices, such as ECG lead-1 and lead-2 sensors, wristwatches (temperature and jogging sensors), and ankle magnetometer sensors. Each sensor can generate real-time data and offload it to the proximity laboratories for processing.
- The proposed SFTS is more efficient regarding security, fault-tolerant, and resource scalability.
- We integrated the federated learning scheme, where training and testing are determined based on a convolutional neural network (CNN). The aggregated node executes all tasks based on their given constraints.

The paper consists of the following parts. The goals of the previous studies for IoT healthcare in fog cloud models were discussed in the related work. The problem architecture shows all components of the architecture. The proposed algorithm part shows how to solve the problem in different steps. The experimental part shows the simulation configuration and simulation results. In conclusion, the results and future direction of the work were looked at in light of the new limits.

2. Related work

Digital twin technology in IIoHT systems has achieved many achievements in the healthcare domain. Different healthcare applications, such as disease prediction, secure data offloading, mobile medicine, and IoT healthcare, are widely integrated with DT technology. Further studies solved the different healthcare issues with additional constraints, as shown in Table 1. These studies (Lv, 2023; Leng et al., 2022; Leong et al., 2021; Chi et al., 2022; Khoso et al., 2021; Younan et al., 2020) discussed Industry 5.0, a new information technology revolution that transforms traditional healthcare applications, architectures, and systems into digital and automated forms. In Industry 5.0, many emerging technologies, such as DT, edge computing, machine learning, cloud computing, and blockchain technologies, are integrated with healthcare to make it more robust and efficient. However, these studies only discussed DT's information flow and advantages with Industry 5.0 for healthcare. Therefore, methods and systems are to be developed based on the given prototypes in these studies.

This study (Khan et al., 2022) suggested DT-enabled fog cloud solutions for different industry applications. For instance, healthcare in intelligent cities, pharmaceutical medicine supply chains, etc. In detail, this paper discussed machine learning, edge computing, and IoT healthcare sensor-enabled DT industry architectures and resource replicas. Different offloading and resource allocation (RA) strategies based on machine learning for edge and cloud computing are listed with their constraints. However, this work is more

Table 1
Existing IIoHT frameworks based on digital twin.

Study	Proposed	Indus.App.	Gap Analysis
(Khan et al., 2022)2022	DT-Industries.	IoT-Health	Resource Replica
(Elayan et al., 2021; Ghita et al., 2020)2021	DT-Context-Algo.	IoT-Health	Resource Scalability
(Volkov et al., 2021)2021	DT-Mobile	IoT Medicine	Resource Scarcity
(Haleem et al., 2023)2021	DT-Healthcare	IoT Health	Feature and Service
(Azzaoui et al., 2021; Jimenez et al., 2020; Akash and Ferdous, 2022; el Azzaoui et al., 2020; Zhang et al., 2020)2021	DT-Blockchain,Security	IoT Health	PoW,Methods
(Nguyen et al., 2022; Darvishi et al., 2021; Alshathri et al., 2023)2020–2022	DT-Fault-Detection	IoT Health	Backup,Checkpointing
(Rieke et al., 2020; Xu et al., 2021)2020–2021	Federated	IoT Health	Training
Proposed Work	DT-Federated Constraints	IoT Health	Fault,Security,RA

general. So far, security, fault tolerance, and other issues will be solved in the discussed solutions.

A healthcare DT-enabled context-aware IoT fog cloud solution has been suggested by this study (Elayan et al., 2021). For prediction, the electrocardiogram (ECG) non-invasive technology data was offloaded to proximity clinical servers. The healthcare clinics are integrated with the digital twins, where homogeneous nodes can share their data assets and virtual server copies. However, this work only focused on limited IoT data services with fixed nodes for healthcare contexts. To improve the efficiency of IoT healthcare context-aware *DT*, a distributed intelligent geospatial system based on cloud services is suggested in Ghita et al. (2020). This system offered distributed context-aware IoT healthcare services based on cloud computing. There is no issue of resource scalability in the work. However, due to the many users of this technology, the storage and processing costs become higher for the service providers. Security scarcity is also a challenging task for this technology.

However, prior studies focused on the context of IoT healthcare services, where data is offloaded based on non-invasive ECG sensors to cloud computing. For the offloaded data, the mobile medicine system based on DT is introduced in these studies (Volkov et al., 2021; Haleem et al., 2023). This work combined different pharmaceutical companies and collected the server data assets. However, the work could be more secure and fault-tolerant. During simulation results, failure of tasks during offloading and scheduling was seen. However, these studies (Azzaoui et al., 2021; Jimenez et al., 2020; Akash and Ferdous, 2022; el Azzaoui et al., 2020; Zhang et al., 2020) suggested blockchain and secure algorithm-based solutions solve the security limitations of prior studies in DT IoT healthcare. Public blockchain technology (Azzaoui et al., 2021) implemented with the DT, where different blocks can share virtual data assets. The main advantage is that the transactional nodes do not need to process and validate previous transactions to avoid delays and resource consumption in fog cloud networks. However, public blockchain technology with DT shares data with homogeneous nodes. Therefore, it cannot be used with heterogeneous nodes in IoT healthcare domains. The IoT healthcare cyberspace (Jimenez et al., 2020; Akash and Ferdous, 2022; el Azzaoui et al., 2020; Zhang et al., 2020), such as the cyber-physical system, is integrated with digital twin technology. DT integrated with mobile and fog cloud networks, securely sharing different data types.

These studies (Nguyen et al., 2022; Darvishi et al., 2021; Alshathri et al., 2023) suggested DT-enabled fault-tolerant techniques such as primary backup and checkpointing on fog and cloud networks for IoT healthcare applications. These studies focused on the compile time failure of services, tasks, resources, and scheduling for the assigned tasks to the fog and cloud networks. The primary backup is integrated into the different fog and cloud networks as virtual servers, where task checkpointing techniques are implemented. These studies (Rieke et al., 2020; Xu et al., 2021) federated learning enabled solutions for distributed healthcare systems. However, the proposed frameworks only support fixed nodes and incur the resource failure of nodes during training and testing in networks.

To the best of our knowledge, SFTS-enabled IIoHT is the new solution. The main reason is that the existing security mechanisms provided by these studies (Azzaoui et al., 2021; Jimenez et al., 2020; Akash and Ferdous, 2022; el Azzaoui et al., 2020; Zhang et al., 2020) in digital twin-enabled fog cloud are only supported on rich resource nodes. Therefore, mobile devices can not integrate those models. The existing digital twin enabled IIoHT considered the homogeneous environment for data replication and execution. However, in our case, we have different fog and cloud nodes. Therefore, federated learning-enabled security and privacy are the new contributions to DT-enabled IIoHT in mobile fog cloud networks.

3. Proposed IIoHT framework

The study presents an IIoHT framework based on digital twin and federated fog-cloud models, as Fig. 1 illustrates. As shown in Fig. 1, we can call architecture to the proposed framework. We designed the framework based on two core technologies: federated learning and *DT*. Federated learning allows different hospitals to train, validate, and securely share their private data. The digital twin is the backbone of the system. Heterogeneous copies of servers offer the same services, like storage, resources, and runtime environments for executing programs. So, our goal is to process healthcare sensor data in the shortest amount of time while remaining secure and efficient. The proposed SFTS consisted of different schemes such as fault-tolerant, security, local processing, offloading convolutional neural networks (CNN), and aggregated methods. The healthcare sensor could be abnormal, so the system could not be slow or fail during a patient's critical condition. The study implemented different healthcare sensors and monitored their healthcare during daily activities. Wearable sensors such as ECG (lead-I and lead-II), wrist-watches, and ankle sensors generate data for mobile devices. Furthermore, mobile devices offload sensory data to nearby hospitals for processing.

The healthcare tasks are mobile healthcare functions and monitoring and offloading the sensory data for some purpose. We monitored that each user or subject performed different daily activities at different intervals. We monitor the users' healthcare based on the generated data from sensors to the system. All these sensors are connected to the hospitals via different communication channels, such as wireless and mobile networks. We implemented two main technologies, digital twin and federated learning, in the distributed fog cloud networks. All the local servers of the hospitals are implemented at the radio network, and the centralized cloud is located at the infrastructure level. We trained and validated offloaded data models at the local hospital networks based on machine learning training models and integrated their weights

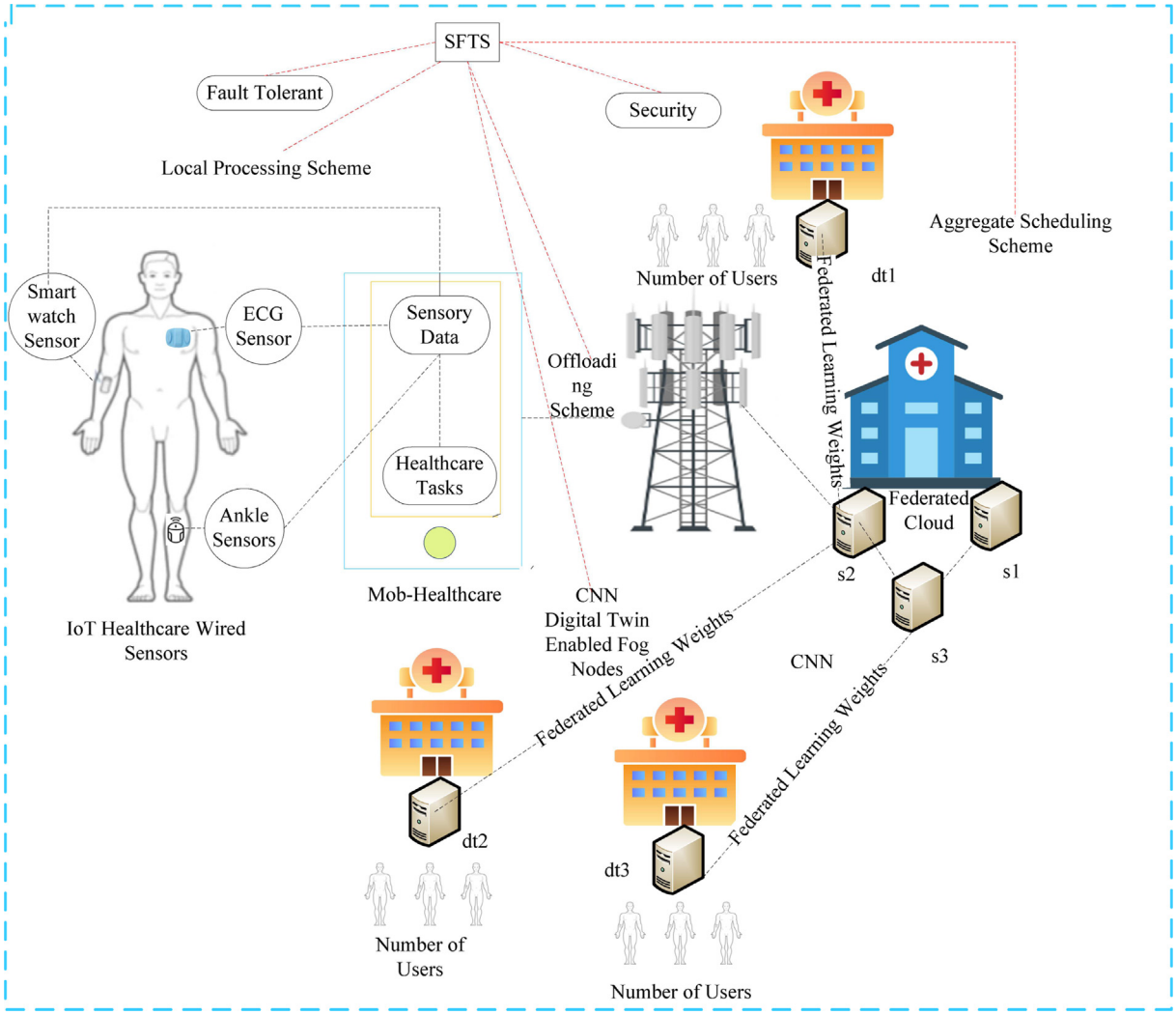

Fig. 1. Secure-fault-tolerant efficient industrial internet of healthcare things framework based on digital twin federated fog-cloud networks.

based on federated learning. All the local fog servers are assumed to be the main server's digital twin. The goal is to control security issues, failure issues, and delay issues during the processing of sensory tasks based on given constraints. Table 2 shows the notations and abbreviations of different terms and mathematical models.

We consider the D number of sensor data with the different number of features F. The features of the data d are $f1$ temperature, $f2$ ECG signal, $f3$ heartbeat speed, $f4$ blood pressure, $f5$ ankle direction, and others. Therefore, we formulated them in the following way. The study considers the T number of tasks. Each task t consisted of data d and features F, deadline d_t, processing status t_s as green annotation shown in Fig. 1 All the healthcare tasks are performed on the user or subject devices as well as fog and cloud nodes. It depends upon the availability of computing resources. Before offloading, we consider the M number of mobile devices. Each mobile m has processing and computing capability represented by ϵ_m and ζ_m. The study considered the S number of healthcare cloud servers and DT number of digital-twin-enabled fog nodes implemented at the radio networks. All the users or subjects can access any server, whether it is a centralized federated cloud or digital-twin-enabled local fog nodes for the services. All the digital-twin-enabled fog nodes and federated cloud servers

Table 2
Abbreviations & notations and description.

Notations and Abbreviations	Description
IoT	Internet of Things
IIoT	Industrial Internet of Things
IIoHT	Industrial Internet of Healthcare Things
IoHT	Internet of Healthcare Things
RA	Resource Allocation
DT	Digital Twin
ECG	electrocardiogram
Mob-Healthcare	Mobile Healthcare
FL	Federated Learning
D	Number of sensor data
F	Number of features
d, f	Particular data and feature
T	Total number of tasks
M	Number of mobile devices
t, m	Particular mobile device and task
ϵ_m, ζ_m	Mobile resource and speed
S	Number of healthcare cloud servers
DT	Digital twin number of fog servers
dt, s	Particular fog node and cloud server
$\epsilon_{dt}, \zeta_{dt}$	fog cloud resource and speed
ϵ_s, ζ_s	Cloud server resource and speed

are homogeneous in execution runtime. However, they are heterogeneous in resource capability. Therefore, each federated resource has ϵ_s resource capability and ζ_s computing speed capability. Similarly, all fog nodes have ϵ_{dt} and ζ_{dt} computing resource and speed, respectively. We designed the mathematical model based on the assignment problem on mobile fog cloud networks. We determined the local sensory processing time of tasks as follows.

$$L_t^e = \sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{d=1}^{D}\frac{d}{\zeta_m} + Encryption. \quad (1)$$

Eq. (1) designed based on mobile fog cloud assignment problem (Daigneault and St-Hilaire, 2021). Eq. (1) analyzes and monitors sensory data's local processing time for specific tasks on mobile devices. In this equation, L_t^e represents the execution time of all tasks, where L is the variable that stores the execution time of all tasks. Furthermore, the inclusion of $t \in T$ indicates that all tasks, from start to end, should be executed after assigning them to their respective computing nodes. The variable e signifies the execution of all tasks on different computing nodes. All the tasks are encrypted before offloading to any server for processing. Therefore, the variable *Encryption* determines the encryption and decryption of all tasks among different computing nodes. The offloading transmission time of the sensory data is calculated as below.

$$C_t^e = \sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{d=1}^{D}\frac{d}{upload} + \frac{d}{download}. \quad (2)$$

Eq. (2) determines communication offloading where tasks are offloaded from local devices to computing nodes. We designed this Eq. (2) based on communication offloading based on the same network rule (Kim, 2020). Eq. (2) determines the data's pre-determined upload and download transmission times before and after processing. On the other hand, C_t^e represents the offloading and downloading of task data from local sensors to the computing servers for processing. The variable C holds the communication time of all tasks from local sensors to computing servers during offloading and downloading results. Furthermore, the inclusion of $t \in T$ indicates that all tasks, from start to end, offload their data and download their results from the servers. The scheduling time on the cloud servers is determined as follows.

$$Cloud_t^e = \sum_{s=1}^{S}\sum_{t=1}^{T}\sum_{d=1}^{D}\frac{d}{\zeta_s} + Encryption \quad (3)$$

We designed Eq. (3) based on cloud scheduling on different computing servers (Panda et al., 2022). Eq. (3) analyzed and monitored the cloud processing time of sensory data for particular tasks. The variable, e.g., $Cloud_t^e$ determines the execution time of all tasks on cloud computing. The digital-twin-enabled nodes have the following processing time for all tasks. Furthermore, the variable of $t \in T$ indicates that all tasks, from start to end, are executed on the cloud servers. All the tasks must be decrypted before starting any processing. Therefore, the variable *Encryption* determines the encryption and decryption of all tasks among different computing nodes. However, after execution, all tasks must be encrypted to share another cloud server for storage. We scheduled all offloaded tasks on digital-twin-enabled fog cloud networks, designed based on digital twin fog cloud rules (Alaasam et al., 2020). We determined the processing time based on twin digital twin fog servers in our work and defined it in the following way.

$$F_t^e = \sum_{dt=1}^{DT}\sum_{s=1}^{S}\sum_{t=1}^{T}\sum_{d=1}^{D}\frac{d}{\zeta_d t} + Encryption. \quad (4)$$

Eq. (4) is designed based on digital twin fog cloud rules (Alaasam et al., 2020). Eq. (4) aims to determine digital twin fog nodes processing time of offloaded sensory data for particular tasks. We implemented the digital twin mechanism of the fog nodes, where F_t^e shows that all tasks are executed on digital-twin-enabled fog nodes. All the fog nodes are represented by $d \in D$, and the variable of $t \in T$ indicates that all tasks, from start to end, are executed on the fog nodes from the scheduler. To present a federated learning approach, we consider the different heterogeneous nodes for data sharing and execution in our work. All the tasks must be decrypted before starting any processing. Therefore, the variable *Encryption* determines the encryption and decryption of all tasks among different computing nodes. However, after execution, all tasks must be encrypted to share another fog server for storage. Therefore, we maintain the data security of tasks on different is determined in the following way.

$$Encryption = \sum_{m=1}^{M}\sum_{d=1}^{D} Enc(d, AES, publickey) + Dec(Enc \sim d, privatekey). \quad (5)$$

Eq. (5) designed based on advanced encryption standard security rule (Dharangan et al., 2022). Eq. (5) determines the encryption and decryption of all task data on different nodes. For example, each node encrypts task data based on a public key using the Advanced Encryption Standard (AES) (Dharangan et al., 2022). Then, the nodes decrypt the task data using a private key. We implemented AES-256 with multiple rounds and specific characteristics such as round substitution and column replacement. This equation, for instance, $Encryption = \sum_{d=1}^{D} Enc(d, AES, \text{publickey}) + Dec(Enc \sim d, \text{privatekey})$, illustrates that all nodes must encrypt and decrypt the data during sharing and execution in the network. Overall, we determined the total processing time of tasks based on minimization optimization enabled on mobile fog cloud rule (Lakhan et al., 2022).

$$\min Total = L_t^e + C_t^e + Cloud_t^e + F_t^e. \quad (6)$$

Eq. (6) is designed based on the mobile fog cloud scheduling rule with the minimization objective with the constraints (Lakhan et al., 2022). Eq. (6) determines the total time of all tasks on different nodes with different features. We denoted the total time as variable *Total* and determined the execution time with the unit minutes. The variable *Total* is an array that stores the execution time on the cloud, the communication time during offloading and downloading, and the processing time on fog nodes for all tasks. We calculate the total time using the equation $Total = L_t^e + C_t^e + Cloud_t^e + F_t^e$. Therefore, individual times impact the total processing time in our architecture.

4. Proposed SFTS algorithm methodology

The study presents the SFTS algorithm methodology, which consists of different sub-schemes. The process flow of these sub-schemes has different connections, as shown in Fig. 2. The SFTS algorithm starts with all parameter constraints such as M, DT, D, S, T. The distinct parameters have already been explained in Table 2. The local processing scheme initiates the healthcare application tasks on local devices. All the sensors are connected to mobile devices, so the sensor data is generated only on local machines. Each local server encrypts the data before offloading and receives decryption results from the digital-twin-enabled fog and cloud servers. Offloading is a communication scheme that transfers generated data from local mobile devices to available fog nodes for further processing. We train the generated data on fog nodes based on data, resources, computation time, and deadline. We apply Convolutional Neural Networks (CNN) to the local training data, and federated learning (Rieke et al., 2020) integrates the local training data into the aggregated node for final results.

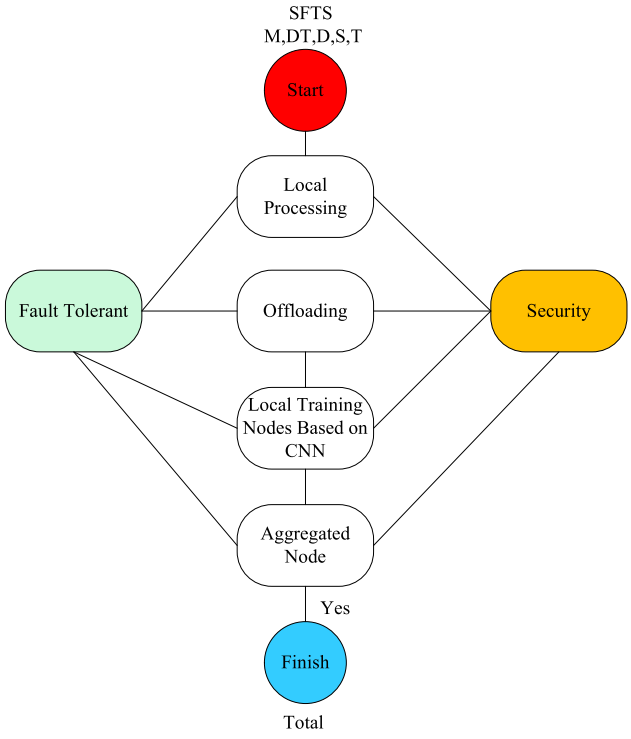

Fig. 2. Proposed algorithm flow diagram.

The security and fault-tolerant schemes are centralized and connect all nodes, enabling them to acknowledge each other about security issues in the IIoHT framework.

Security and fault tolerance are the key challenges in the IIoHT healthcare architecture. Therefore, the study proposed the SFTS framework that consists of different sub-schemes. SFTS framework consisted of different schemes as shown in Algorithm 1.

Algorithm 1. SFTS Algorithm Framework

Different levels exist in Algorithm 1 for generating and processing sensors based on their constraints. Level 1 is the scheme where sensor data are processed based on available resources and offloaded to the available fog and cloud nodes. Level 2 is the communication channel that transmits the data without security and failure issues. Level 3 local federated learning training and testing models. Level 4 is the aggregated node processing. Level 5 handles the fault-tolerant mechanism of the study. We defined the all steps of Algorithm 1, where the flow starts from local processing to fog and cloud nodes in the following way.

- The local processing scheme performs based on Eq. (1). All the local nodes are sources of data generation. To ensure security and privacy, all the task data is encrypted before being sent to the fog and cloud servers for processing, as shown in Level 1.
- Offloading is the mechanism where the algorithm checks if the communication network and computing resources are available; it allows local devices to offload data to the servers. This algorithm's main efficiency is providing a seamless environment when all the network's communication channels and computing node resources are available.
- In Level 3, all the tasks are offloaded to available fog nodes for processing. The fog nodes are integrated with the digital twin, where resources, task updates, and execution status training and testing are shared with the cloud nodes for aggregation. The main efficiency of the digital twin is its ability to enable fog nodes to provide resource availability at the edge layer. These fog nodes act as replicas of cloud computing servers, having the same runtime environment and interoperability within the network.
- The aggregation takes place in the cloud computing nodes, as shown in level 4, where all fog nodes share their task execution status and resources and store the final results of the tasks. This sharing is done to achieve resource scalability. We devised an aggregation mechanism based on vertical federated learning, where all the fog nodes share metrics such as resources, task status, failure annotations, and trained and tested execution models with the cloud computing servers. This sharing aims to further optimize and improve the nodes' efficiency for all tasks.

Algorithm 1: SFTS Algorithm Framework

Input : M, DT, D, S, T

begin
  Level-1 Local Processing Sensory Data;
  Level-2 Offloading Data;
  Level-3 Digital Twin Fog Nodes;
  Level-4 Aggregated Cloud Nodes Enabled Scheduling;
  Level-5 Fault-Tolerant Mechanism;
  Optimize $Total \sim M, DT, D, S, T$;
End Levels;
return $Total$;

- The fault-tolerant scheme, as shown in Level-5, is an important component implemented on all nodes. The main objective of fault tolerance is to minimize the risk of task failures. It involves training and rescheduling all failed tasks on available computing resources within the given task deadline.

4.1. Local processing sensory data

All the local healthcare sensor data are generated by the different sensors and connected to mobile devices. The local mobile devices analyzed the security based on the following rules.

Algorithm 2. Local Processing Sensory Data

Algorithm 2: Local Processing Sensory Data

Input : D, M, T

1 **begin**
2 **foreach** *(T annotated as local tasks t)* **do**
3 **if** *($L_t^e \sim D \leq \epsilon_m$)* **then**
4 $@t$ local processing based on available resources;
5 **if** *($@t \leftarrow L_t^e \leq d_t$)* **then**
6 Determined the local time based on equation (1);
7 Apply encryption and decryption based on equation (5);
8 Search available resources based on equation (8) after encrypted all tasks;
9 DT, S;
10 Offload to based on Algorithm 3 to fog servers D;
11 End Offloading;
12 End Levels;
13 **return** $Total$;

We present the local processing scheme Algorithm 2 that processed the local process as the IoT sensor healthcare mobile data. We discussed the IoT healthcare data based on mobile devices in the form of a case study, as shown in Fig. 3. For understanding purposes, we consider only three types of sensors. For instance, ECG sensor (lead-I and lead-II), wristwatch sensors (temperature and heart-beat range), and jogging speed and walk count sensor that is integrated into the subject or user's ankle. We apply the wavelet scattering technique (Jean Effil and Rajeswari, 2022) to pre-process data according to features. However, it depends upon the available resources; if the required pre-processing and security resources are sufficient, then the mobile devices are processed locally. Otherwise, this process will offload to the available digital twin fog nodes. The offloading schemes proposed in Algorithm 2 are defined in the following way.

- Initially, all tasks are annotated as local tasks, e.g., $@t \in T$, where local computing nodes (e.g., $m \in M$) execute all tasks to meet the security and privacy requirements of data locally. Therefore, the inputs equal D, M, and T. T is the total number of tasks, D is the sensor data, and M is the set of local computing nodes for processing the sensory data at the local machines.
- In Steps 1 and 2, all the tasks are annotated as local tasks. The main reason is that each task must be initially executed locally. We read the tasks one by one from the task set, as shown in Steps 1–2.
- Step 2 verifies that the local devices have enough to execute sensory data. Therefore, initially, we anticipated resource checking before execution for all tasks.
- In Step 4, the algorithm determines that if the local computing nodes have enough resources, it annotates all tasks as local tasks and starts their execution based on security requirements.
- In Steps 5 to 7, we ensure that the tasks are executed within their deadlines and meet the security requirements. The algorithm allows execution based on Eq. (1) for task execution and security based on Eq. (5).

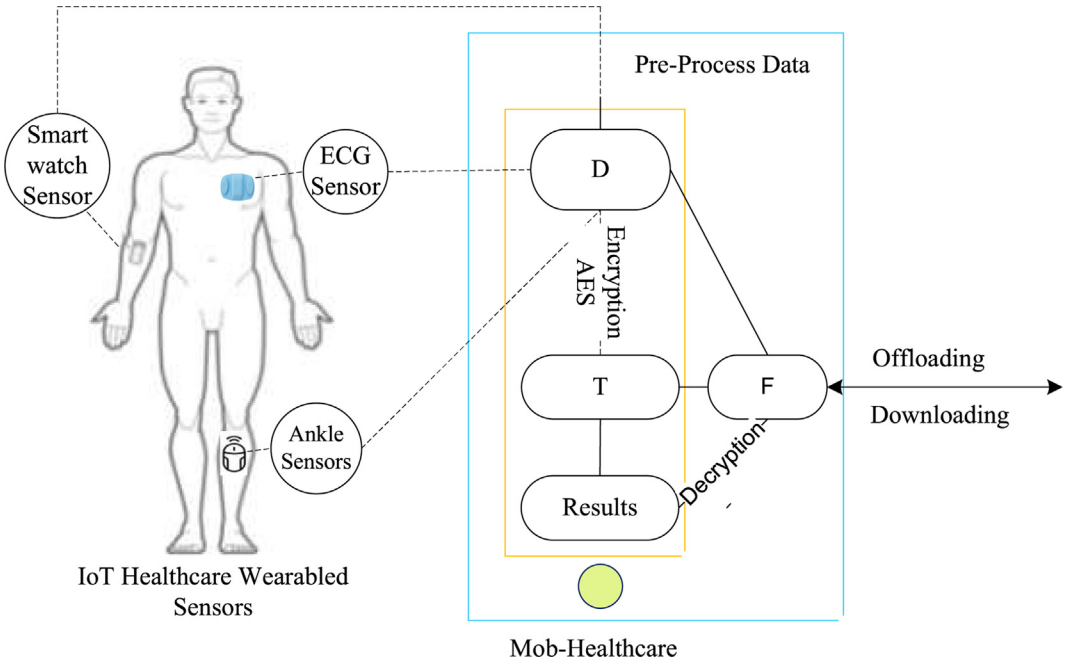

Fig. 3. Local processing on sensor data.

- In Steps 8 to 11 determines that the local processing finished their execution and is looking for offloading for execution based on available resources and wireless communication.
- We determined the available resources based on Eq. (8) before starting any execution and offloading for all tasks.
- The local nodes offload their data to the fog and cloud nodes. After the execution, all nodes send back their results in encrypted form, decrypted by the local device for display.

4.2. Offloading data

Offloading is a process that initiates from mobile devices when they have no resources or tasks that need further execution based on their given thresholds.

Algorithm 3. Offloading Data Scheme

Algorithm 3: Offloading Data Scheme

Input : M, DT, D, S, T

1 **begin**

2 **if** *(Communication.availability==true)* **then**

3 Determined the communication time based on equation (2);

4 $C_t^e = \sum_{m=1}^{M} \sum_{t=1}^{T} \sum_{d=1}^{D} \frac{d}{upload} + \frac{d}{download}$;

5 End Offloading;

6 Acknowledge Generated;

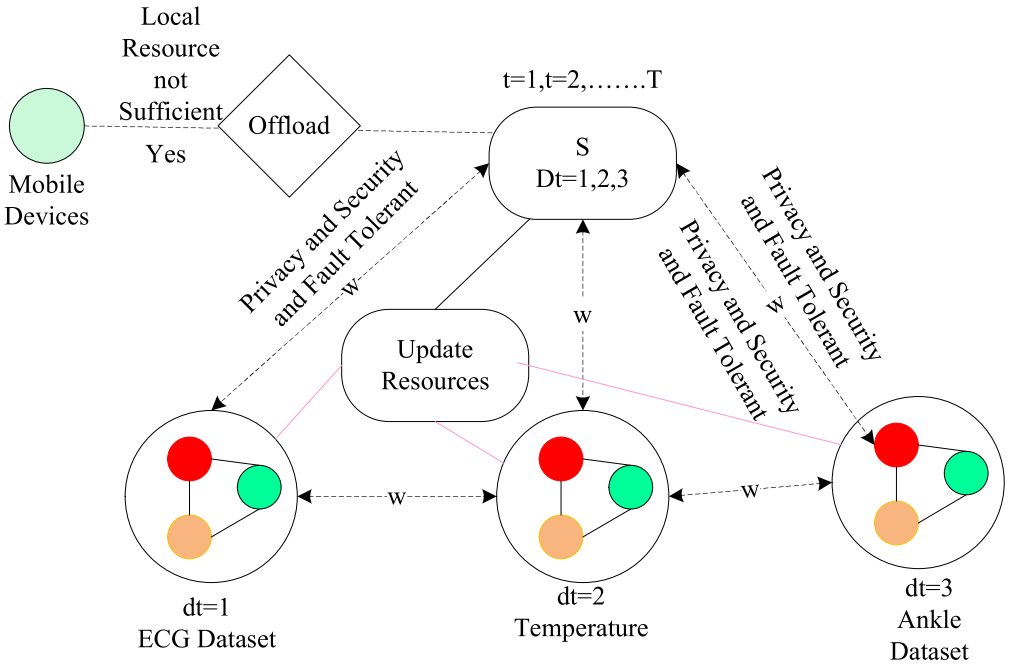

Fig. 4. Federated learning processing on digital twin.

The study designs the offloading scheme Algorithm 3, where all data is offloaded from local mobile devices to the available fog and cloud nodes for processing. Algorithm 3 checks in advance; if the communication channels have a higher capability of offloading, it will offload data to the available fog node for further processing. Otherwise, if the mobile offload finds weaker and insufficient bandwidth, the mobile waits for the network availability and resource availability inside the designed network. We define the steps of Algorithm 3 in the following way.

- In Steps 1–2, the algorithm checks communication availability for offloading data to the connected fog and cloud nodes.
- In Steps 3–4, the algorithm determines the communication time and availability for offloading and downloading data from computing nodes to the local machines.
- In Steps 5–6, determines if the communication channel is established, then the algorithm allows the local processing algorithm to offload all tasks to the available computing nodes for execution.

4.3. Federated learning enabled local training and testing models

The study designed the federated learning and its replica virtual local fog nodes at different radio networks. In our case, we implemented federated learning at different nodes. The main aggregated node is powerful cloud computing, where digital-twin-enabled same virtual copies are integrated at different healthcare clinics as shown in Fig. 4. Each node has weights $W = \{w = 1, dots, .W\}$ that consists of different features (e.g., resource, deadline, training, and testing datasets). All the cloud and fog nodes in federated learning-enabled digital twins cooperate and communicate without overhead issues. Data security and privacy are maintained among nodes based on encryption, decryption rules, and fault-tolerant techniques. We present the federated learning technology-enabled framework for IoT wearable healthcare devices. In our system, federated learning is the complete system based on cloud data servers and its replica fog servers based on digital twin technology. The federated is divided into local training and testing part and aggregated decision parts for IoT healthcare devices. The study considers the *FLM* number of federated learning training servers such as $FLM\{flm \sim dt \sim s1\}$. We trained both replica and aggregated nodes based on resources capability and IoT datasets and represented by $Z = \{z = 1, \ldots, Z\}$. In our study, we considered horizontal federated learning where all fog and aggregated share their datasets but in different sample modes. We determined the federated learning mechanism based on the following equation.

$$\min_{w} f(t) \sum_{flm1}^{F} LM \sum_{dt=1}^{DT} \sum_{s=1}^{S} flm(w|z). \tag{7}$$

We designed the Eq. (7) based on federated learning rules for fog and cloud networks (Lakhan et al., 2021). In Eq. (7) w is the learning weight of federated learning on given input $t \in T$ tasks and trained dataset z. It is the same for all datasets on different nodes. We trained and test the model based on a neural network. We divide the objective function performance into local, fog, and cloud components as shown in Eq. (7).

Algorithm 4: Federated Learning DT Scheme

Input : D, M, T, FLM, S

begin
  $W[\epsilon_{dt}, l, s] \in DT, L, S$;
  if *(offloading.status=true)* **then**
    foreach $(flm \leftarrow d)$ **do**
      Activate Security and Fault Tolerant Analyzers;
      Determined the communication time based on equation (2);
      $\frac{d}{upload} + \frac{d}{download}$;
      Apply encryption and decryption based on equation (5);
      Determined the federated learning among fog and cloud nodes based on equation (7);
      $\min_w \leftarrow Total \sim f(t) \sum_{flm1}^{F} LM \sum_{dt=1}^{DT} \sum_{s=1}^{S} flm(w|z)$;
      Determined the fog nodes execution time based on equation (4);
      $F_t^e = \sum_{dt=1}^{DT} \sum_{t=1}^{T} \sum_{d=1}^{D} \frac{d}{\zeta_d t}$;
      Apply encryption and decryption based on equation (5);
      Determined the cloud nodes execution time based on equation (3);
      $Cloud_t^e = \sum_{s=1}^{S} \sum_{t=1}^{T} \sum_{d=1}^{D} \frac{d}{\zeta_s}$;
      Apply encryption and decryption based on equation (5);
    End Initial Process;
    if $t \leftarrow Total \leq d_t \& t_{status} == 1$ **then**
      Determined the weights and execution based on given datasets;
    End Training and Testing;
  End Processing;
End Main;

Algorithm 4 schedule all tasks based on their given features and quality of service requirements (deadline and total) on the different resources. Algorithm 4 initiated with the different weights, e.g., $W[\epsilon_{dt}, l, s] \in DT, L, S$. We consider the mobile device the thin client where tasks such as annotated@ must be scheduled and executed securely. However, IoT healthcare tasks are generally scheduled in real-time, so due to resource limitations in mobile devices, the mobile engine offloads task workloads to the external fog cloud servers for further processing. All the datasets are trained and tested at the fog nodes individually based on the neural network. Each dataset has different attributes during training and testing on the given resources. All the fog nodes are digital-twin-enabled

virtual servers of the centralized aggregated main cloud. Cloud computing also has different heterogeneous to make the final decision on the tasks. Algorithm 4 ensures that all the tasks must be executed under their given deadlines and optimized the overall *Total* of all tasks. Algorithm 4 determines the local execution time based on Eq. (1, and communication offloading time based on Eq. (2), and cloud and fog nodes computation time based on respective, Eq. (3) and Eq. (4). Furthermore, we analyzed the resource availability of nodes in Algorithm 4 as follows.

$$Resource = \sum_{l=1}^{L}\sum_{dt=1}^{DT}\sum_{s=1}^{S}\epsilon_{dt} \leqslant d \in total \leftarrow t \leftarrow D \leqslant l, dt, s \tag{8}$$

This Eq. (8) is designed based on available resources and meets the deadline rules (Naha et al., 2020) for all tasks on different nodes. Eq. (8) determines the resource capability of nodes before scheduling workloads to them must have higher resources than requested works. We generally divided the tasks into critical and general status. The critical tasks are those tasks that have higher processing delay than the given deadlines and consume much more resources in the mobile fog and cloud nodes, as shown in Fig. 5. The study considers the security risk analyzer and fault tolerant analyzer for verifying tasks and resources' security and fault-tolerant status in designed architecture as shown in Fig. 5. The general tasks are executed normally from submission to execution without any security and fault-tolerant issues. The security risk is raised when no resource is available for encryption and decryption security mechanisms in the shared resources. In our case, all the resources and computational nodes (e.g., mobile, fog, and cloud) are shared, and the public key of the encryption algorithm AES-256; however, each node has its own private key to decrypt task data. The critical tasks are the failure tasks due to a lack of resources in mobile fog and cloud networks. In this paper, the study presents the digital-twin-enabled fault-tolerant scheme (DTFTS). The main purpose of the scheme is to identify handle and predict the failure of resources and task states before and during execution. DTFTS identifies the failure patterns of resources and tasks and trains them based on the model to avoid future future. IoT healthcare systems combine very complex computational components such as mobile devices, fog nodes, and cloud nodes that are heterogeneous in nature. At

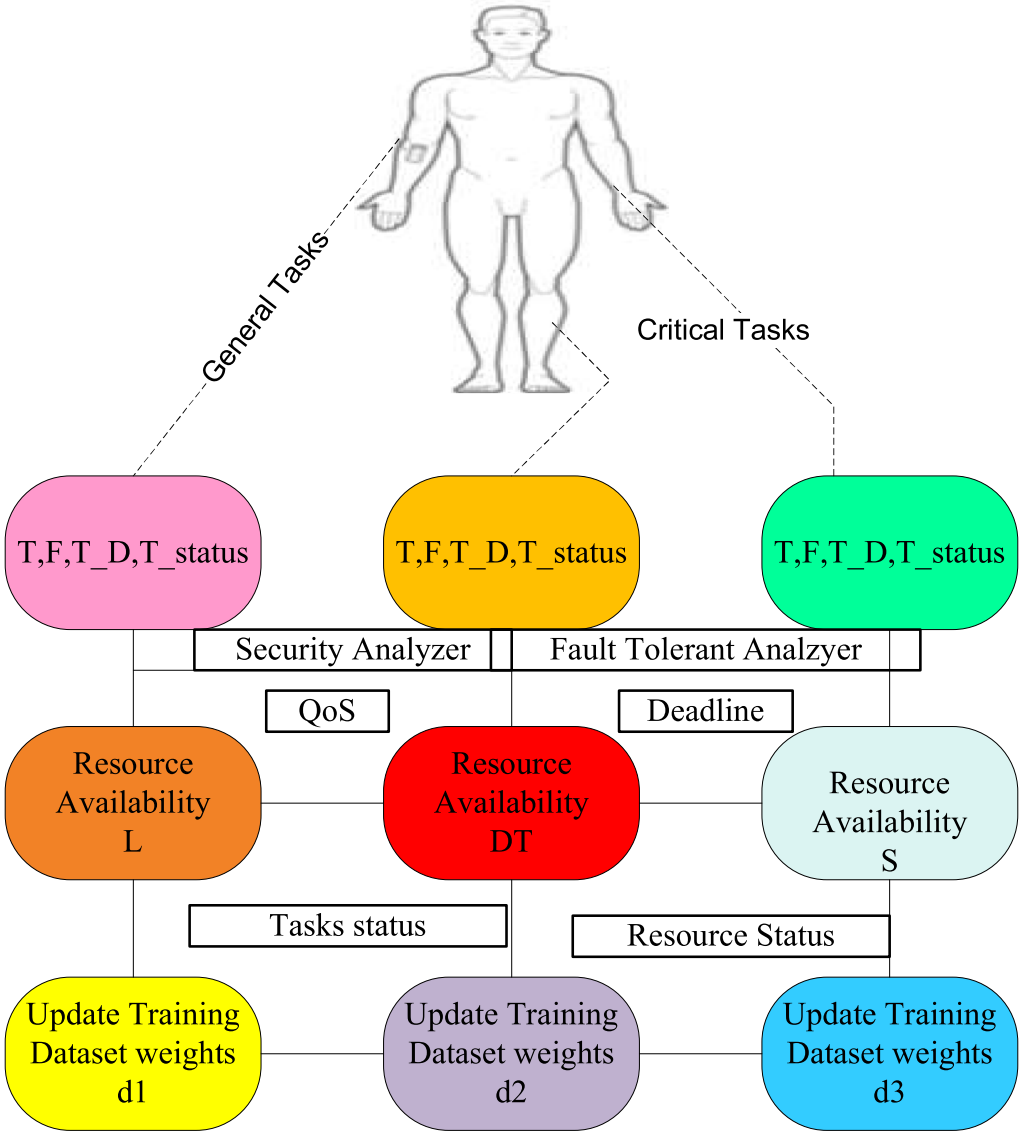

Fig. 5. Resource management in IIoHT healthcare.

the same time, each task has different features, and it may require trained data from different nodes during execution based on the given quality of service requirements. The failure analyzer scheme DTFTS works on different three layers, such as mobile local processing, fog, and cloud processing, as shown at the top lever in Fig. 5 with the different colors. The purple color shows that the security and failure analyzer handles all failure and security risks on the mobile devices $l = 1, \ldots, L \sim \epsilon_l$ based on available resources and acknowledges the fog and cloud nodes for further execution. The cyber yellow and green show that the fog nodes $dt = 1, \ldots, DT \sim \epsilon_{dt}$ and cloud nodes $s = 1, \ldots, S \sim \epsilon_s$, respectively. All the computing nodes work together to maintain the quality of service (quality of service), such as security and fault-tolerant, deadlines, and resources for all IIoHT healthcare tasks. The goal is to maintain task performance and meet all quality of service requirements among different computing nodes. At the second level, we analyze that mobile, fog nodes, and cloud nodes acknowledge each other about the availability of the resources and tasks execution, such as tasks status and resources status in different brown, red, and sky blue nodes. The mobile, fog, and cloud node weights are updated in real-time, and pre-trained and dynamic training is done based on the neural network (convolutional neural network) algorithm (Unnisa et al., 2023).

A convolutional neural network (CNN) algorithm is used in the study to pull out the features of different IIoHT healthcare datasets. The three different datasets (e.g., ECG, temperature, and ankle sensors) as inputs. We have non-linear data. Therefore, we implemented the CNN relu function on the initial phase of the algorithm, as shown in Fig. 6. We extracted the required features F from distinct datasets during processing. We considered that each dataset contained different features of tasks and extracted the maximum number of features from different datasets at the Maxpool layer. Furthermore, the pooling layer classified and differentiated the features among different nodes. All the features are matched according to the algorithm's security, deadline, resources, and fault tolerance. The security and fault-tolerant analyzers' schemes executed all tasks at the softmax layer and got the optimal $Total \leftarrow T$ as shown in Fig. 6. We define the steps of Algorithm 4 in the following way.

- Algorithm 4 takes the input from offloaded data from the local devices. We determine that the federated learning DT scheme has the following metrics, e.g., D, M, T, FLM, S. The federated learning training and learning determines the variable, e.g., FLM.
- In Steps 2–3, all the fog nodes scheduled the offloaded tasks and generated their weights. For example, the weights are the execution status of tasks, deadline meeting, resource availability, and failure of tasks in the network.
- In Steps 4–9, all the fog nodes started execution, where security and fault tolerance are trained based on the analyzer. The analyzer is a method that stores the algorithm's training based on a deep neural network, where deadline, failure, and security metrics enable model training. The digital twin-enabled fog nodes trained their models based on federated learning and shared their weights with cloud computing for further execution.
- In Steps 10–19, all the federated learning enabled fog nodes executed all tasks and trained their models based on deadline, execution status, security, and failure status constraints based on Eq. (4) and Eq. (3). All the tasks are executed on fog nodes with a similar runtime based on a digital twin scheme. Due to resource scarcity, the algorithm offloads all completed functions to cloud computing for storage and further processing. We determined the federated updated weights among fog and cloud based on this formula, e.g., $\min_w \leftarrow Total \sim f(t) \sum_{flm1}^{F} LM \sum_{dt=1}^{DT} \sum_{s=1}^{S} flm(w|z)$ based on Eq. (7). Based on federated learning, all the nodes shared the updated weights with each during the execution of tasks.
- In Steps 18–19, the algorithm determined that if all tasks are executed successfully with the deadline and security requirements without failure. The algorithm terminates after the execution of tasks.
- All the fog and cloud nodes are connected based on a federated learning scheme, sharing their tasks and resources status to the aggregated node in form weights.
- All the weights are updated after some time, as shown in Algorithm 4.
- Algorithm 4 performs efficiency on different nodes, where the tasks' deadline, security, and failure efficiency are met.

4.4. Fault tolerant efficient scheme in digital twin

Real-time data processing is another challenging task in digital healthcare when a person has health problems. In our case, a person or patient performs different activities and relies on remote healthcare services to monitor their health in real-time. But from the provider's point of view, a failure of services could change the current state of healthcare sensors and cause a significant loss. We used fault-tolerant techniques made possible by the digital

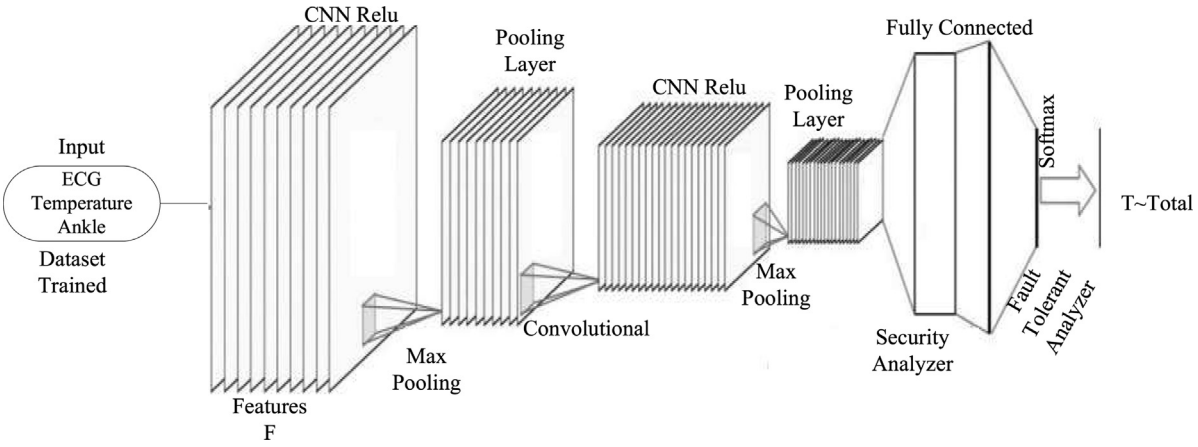

Fig. 6. CNN-enabled training and testing in mobile fog cloud iIoHT healthcare nodes.

twin to deal with service or resource failures. We integrated the digital twin on the service provider side (e.g., hospitals). The digital twin allows the provider to make virtual copies of servers at different layers (e.g., fog and cloud) with the same runtime environment. We put the digital twin servers at various hospitals that offer the same services. Fig. 7 shows a scenario of the fault-tolerant technique based on the digital twin as an execution process for healthcare tasks. Initially, all the tasks, e.g., $t = 1, \ldots, T$, are scheduled based on the particular computing node for the processing. The study presented a fault-tolerant technique based on the checkpointing mechanism (Yang et al., 2022) as shown in Algorithm 5. If the task status changes to 1, it means that the process of the scheduled task will be completed, and it will return the status "processed finished" and send the task results.

Algorithm 5. Fault Tolerant Efficient Scheme in Digital Twin Scheme

Algorithm 5: Fault Tolerant Efficient Scheme in Digital Twin Scheme

Input : DT, M, T

begin
- Initial Schedule;
- **for** $(t = 1 \ to \ T)$ **do**
  - Scheduled on mobile devices;
  - $t \leftarrow m$;
  - **if** $(t \leftarrow m \sim t_{status} == 1)$ **then**
    - All the tasks are successfully offloaded;
  - **else**
    - Search for another device to reschedule;
    - $t \leftarrow m1 \sim t \leftarrow m2$ until $t \leftarrow m2 \in M \sim t_{status} == 1$;
  - **else if** $(t \leftarrow dt \sim t_{status} == 1)$ **then**
    - All the tasks are successfully executed;
  - **else**
    - Search for another digital twin server to reschedule;
    - $t \leftarrow dt1 \sim t \leftarrow dt2$ until $t \leftarrow dt2 \in DT \sim t_{status} == 1$;
    - **if** $(t \leftarrow dt2 \sim t_{status} == 1 \leq d_t)$ **then**
      - All tasks rescheduled from the point of failure in completion %;
      - $t \leftarrow s1 \sim t \leftarrow s2$ until $t \leftarrow dt2 \in DT \sim t_{status} == 1$;
    - **else**
      - Restart tasks from initial scheduling;
    - Failure Tasks Scheduled;
  - End New Start;
- Scheduled all tasks;

End Main;

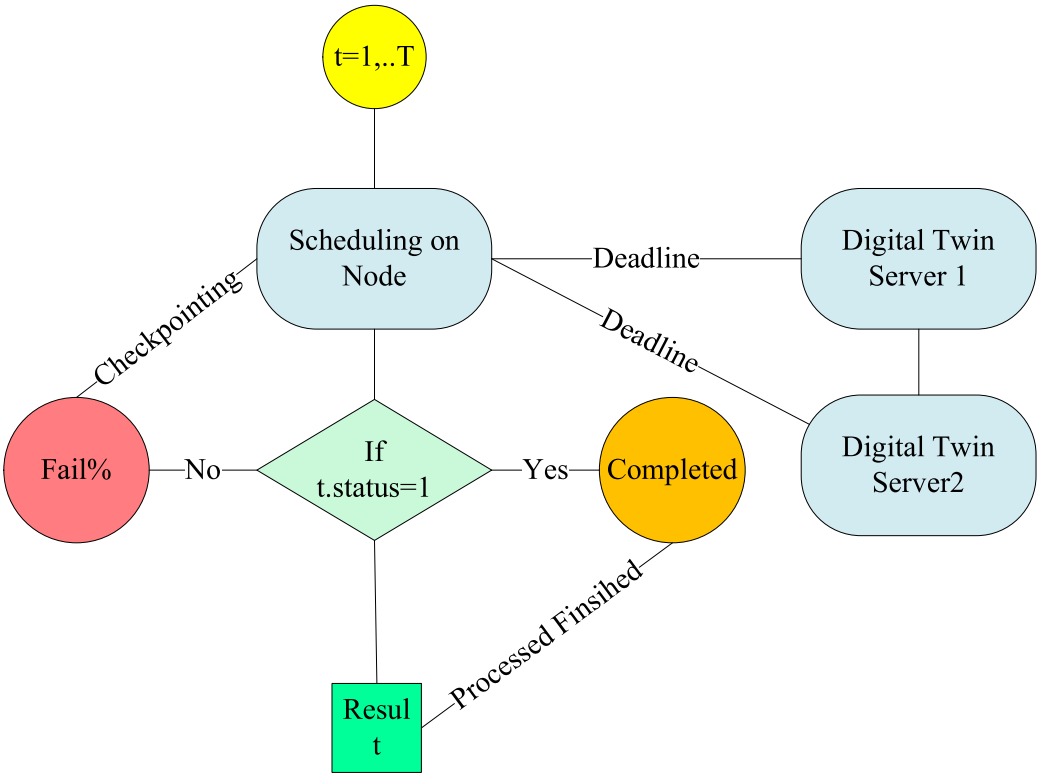

Fig. 7. Fault-tolerant efficiency based on digital twin scenario.

Algorithm 5 considers the failure of using mobile devices and digital twin servers before offloading and after processing, as shown in Fig. 7. If a task t fails due to a sensor's fault or any other issue, it reschedules from the mobile device to another. If the tasks failed on the server side, we applied the checkpointing technique and rescheduled tasks from the point of failure % under their given deadlines, as shown in Algorithm 5. All the failure tasks with the status 0 or no will be rescheduled from the point of failure on the available digital twin servers. During execution, all tasks have two statuses, such as 1 and 0. If the status converts from 1 to 0, the scheduler will try to reschedule it from the point of failure and search for a similar server in the digital twin with similar services. The primary role of digital twin technology is to control the load balancing issue and the service restart problem and to search for similar services under the given deadlines. The digital twin-enabled infrastructure helps a lot to meet the deadlines for tasks. The deadline is critical because if a person uses these healthcare services and encounters some health issues or sensors that generate abnormal data. Therefore, processing anomalous data within the deadline is necessary, as shown in Algorithm 5.

The process shown in Fig. 7 is more effective, where task execution is monitored in real-time, and deadlines of tasks are necessary to meet. The main reason is that abnormal data could be generated randomly. Therefore, the system must be adaptive and control all failures, security, and deadline during their execution at different hospitals. We held the security and failure based on digital twin and federated learning approaches for real-time healthcare applications. Algorithm 5 meets all the requirements of the healthcare tasks in any condition while performing their activities. The fault-tolerant based on digital twin scheme enabled Algorithm 5 has the following steps.

- In Steps 1–2, Algorithm 5 monitors the task execution status on different computing during scheduling in the networks (e.g., mobile, fog, and cloud nodes).
- In Steps 3–5, the algorithm monitors those tasks scheduled on mobile devices. If tasks become 1, it shows that the execution of tasks is being processed optimally. However, the task's status convert from 1 to 0, so the particular tasks are stuck in their executions.
- In Steps 6–24, the algorithm monitors the tasks scheduled on fog and cloud nodes. If the tasks become 1, it indicates that the execution of tasks is being processed optimally. However, if the task status changes from 1 to 0, the particular tasks are stuck during their execution and are marked as failed. If the tasks fail due to resource or security reasons, the scheduler attempts to reschedule the tasks from the point of failure to another node. The algorithm searches for optimal nodes among fog and cloud for rescheduling from the point of failure. If the tasks cannot be successfully recovered, all tasks are rescheduled from scratch on mobile devices. After the local execution, the tasks are scheduled on different computing nodes again until the execution of tasks is completed within their deadlines and for security purposes in the network.

4.5. Time complexity and limitation of SFTS

The proposed SFTS amalgamates different schemes, such as scheduling, federated learning, offloading, and digital twin. Therefore, the time complexity is determined while IoT tasks execute on the other computing nodes. The nodes are mobile devices, fog, and cloud servers. We analyze the time complexity of proposed SFTS algorithms utilized in IoT and fog computing context scheduling on different computing. Our time complexity is total delays as we determined in Eq. (6) and denoted as *Total* for all IoT tasks. In our architecture, we executed IoT tasks on different nodes. Therefore, we divided the time complexity into different computing nodes. The local devices' time complexity is determined in the following way, e.g., $n(n \times n)$. The local devices are initial devices where IoT tasks are initiated at the start of their processing in

our architecture. It is partially polynomial during local processing time and offloading before final execution at servers. The tasks are executed on different fog and cloud nodes. The federated learning computing time is as follows: e.g., $n(n \times n)^4$. The power n^4 shows three digital twin servers and one powerful centralized computing machine during processing in different hospitals. Therefore, the total time complexity, including security, offloading, scheduling, and fault-tolerant, becomes $n(n \times n) + n(n \times n)^4$.

There are areas for improvement of SFTS in the time complexity of different constraints. For instance, the waiting time for mobile devices for offloading has yet to be determined. Therefore, IoT context tasks could suffer deadlines and performances due to long wait times. The federated learning fog and cloud nodes still have load-balancing issues. Therefore, related balancing issues, fog, and cloud nodes give a longer wait time. It impacts the time complexity of IoT fog cloud tasks in our architecture.

5. Performance evaluation and simulator

The study designed an IIoHT healthcare simulation configuration environment based on software and hardware requirements, as shown in Table 3. The users or subject environment is considered the local devices with the X86 operating system for Android and iPhone, where open-source Azure cloud service is integrated into Table 3. We collected sensor data (e.g., ECG lead-1, lead-2, Ankle, and Wrist Watch). We set the parameter and hyperparameter of CNN as mentioned in Table 3. The study designed the experiment based on standard deviation, mean, and median values on the dataset numerical values. IIoHT healthcare consists of local devices, communication channels, fog nodes, and cloud computing. The simulator consisted of different parameters as shown in Table 3. The simulated design is based on available libraries such as the Android developing tool and fog cloud virtualization tools. Table 4 shows the configuration of the local mobile devices during simulation for healthcare sensor data. Table 5 shows the configuration of the digital-twin-enabled fog nodes during simulation for healthcare sensor data. Table 6 shows the configuration of the cloud nodes during simulation for healthcare sensor data.

5.1. IIoHT sensor datasets

The study exploited the public healthcare sensors; datasets are publicly available on the following URL. https://github.com/ABDULLAH-RAZA/IIoT-healthcare-Sensors-Data. The datasets have different features of tasks such as ECG Lead-1, Lead-2, Ankle, BP (blood pressure), Temp (Temperature), HB (Heartbeats), CH (Cholesterol) speed, location, real-time, stayed Activity, and Users. The more definition and attributes datasets are available on the given link.

Table 3
Experiment parameter and values.

Parameter	Values
Convolutional Neural Network layers	16 layers
Pooling size	6
Functions	8
CNN layers	16× 16
Language	IoT C
Cloud	Azure
ECG Lead-1,Lead-2	Rhythm Strip
Ankle	Elastic straps
Wrist Watch	Tempreture
Maxpool	100 GB
Communication	GPS
Speed	5G
Wifi	56mbps
upload bandwidth	60 MHz
Download bandwidth	60 MHz

Table 4
Mobile devices configurations.

Parameter	Values
Operating System Mobile Devices	Android-X86
$l1$	samsung s23 64 GB, 4 GB RAM
$l2$	Iphone 11 128 GB, 8 GB RAM
$l3$	Samsung Galaxy Ax3 series 150 GB, 10 GB RAM
L	3

Table 5
Digital twin fog nodes configurations.

Parameter	Description
Operating System Computing Nodes	Android-X86
$dt1$	Virtual Android-X86, 300 GB, 32 GB RAM
$dt2$	Virtual Android-X86, 500 GB, 48 GB RAM
$dt3$	Virtual Android-X86, 800 GB, 56 GB RAM
DT	3

Table 6
Cloud computing configurations.

Parameter	Values
Operating System Cloud Computing	Android-X86
$s1$	Android-X86, 10000 GB, 100 GB RAM
$s2$	Android-X86, 20000 GB, 200 GB RAM
$s3$	Android-X86, 50000 GB, 300 GB RAM
Total number of nodes$S$	3 Heterogeneous nodes

5.2. Result analysis and discussion

IoT and Fog Cloud Time Complexity The study conducted the experiments based on collected IIoHT sensor data as mentioned above in the URL. We implemented the baseline studies that are closely related to our study. For instance, IoT Without DT-Federated (Rieke et al., 2020), IoT With DT-Federated (Xu et al., 2021), IoT healthcare security and fault tolerant in Fog Cloud (Azzaoui et al., 2021; Jimenez et al., 2020; Akash and Ferdous, 2022; el Azzaoui et al., 2020; Zhang et al., 2020) are the studies implemented in the experimental environment. We represent the total time of tasks in minutes at different mobile, fog, and cloud computing nodes. All healthcare tasks are executed on different nodes. Therefore, the total variable combination of local, fog, and cloud processing times is *Total*, calculated per minute for all tasks as shown in Eq. (6). Fig. 8 illustrates the performance of 26 distinct healthcare tasks conducted in an experimental environment. We analyze security and failure risks within the IoT healthcare domain. Fig. 8 depicts varying delays experienced at different computing nodes: mobile, communication, fog, and cloud during the security and failure process of tasks. In the initial experiment, we analyze the security delay and failure delay with the proposed schemes. The y-axis shows the total as the objective function, determined in minutes. It starts from 0 to 18 min. The x-axis indicates the number of healthcare tasks ranging from 1 to 26. In our proposed work, we can not execute all tasks locally. Therefore, the local devices can initiate tasks and offload to the fog and cloud through communication networks. The mobile devices incurred the initial

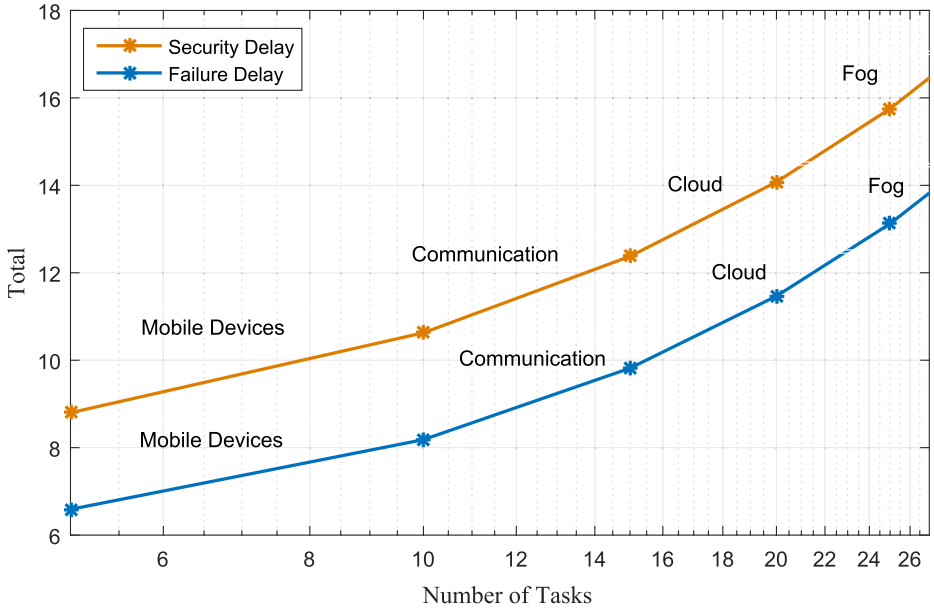

Fig. 8. Secure and fault-tolerant total processing time of all tasks on different computing nodes.

failure of connection or availability of communication networks and remote nodes. For instance, we are offloading to the remote computing nodes such as fog and cloud, incurring 14-min delays. All the nodes are executing tasks in parallel ways. Therefore, we need to minimize the failure delays for all reasons to meet the requirements of tasks in higher with the deadline constraints. Due to the risk to the healthcare of humans, we set deadlines for all tasks and ensure all tasks must be executed under their deadline. We denoted the brown line as the security task validation in the form of delay on different computing nodes. We represented the security delays on different computing nodes, such as mobile devices, communications, fog, and cloud nodes. The *Total* is the objective function that combines various delays. Fig. 8 shows that all the tasks, e.g., 26 tasks as shown in the x-axis, incurred eighteen minutes (18) delays at different nodes as shown in the Y-axis. These delays include encryption decryption and validation of tasks at nodes. Therefore, it is necessary to process all tasks on different nodes with fewer delays according to given tasks.

In our scenario, we consider the random number of healthcare tasks and increase their sizes in the experiment. We implemented data pre-processing strategies, such as wavelet scattering and CNN, for feature extraction in the IIoHT framework. We evaluated the performances of schemes in the experiment. Fig. 9 analyzes the performance of tasks with the different delays as a total delay. The variable *Total* is the sum of processing delays determined in minutes. The total delay becomes higher and higher during the recovery of the security and failure risks for all tasks with the higher number of tasks in different mobile fog and cloud computing nodes. The y-axis represents the delays of tasks as annotated as *Total*, and the y-axis shows the range of tasks with random numbers, e.g., 30. We recover the security failure issues on different computing nodes. For instance, mobile devices performed on the local devices and communication networks identified the original data in encrypted form. However, if there are some issues in the wireless communication and require a recovery time, the offloaded tasks wait until connection recovery from security failure. It is similar to the failure delay recovery, which takes delays for all tasks on different computing nodes. We denoted the number of tasks from 0 to 5 on mobile devices with the 6-min delay with the security recovery. At the same time, 9 min of delay with the failure of tasks during recovery on mobile devices. We offloaded tasks through wireless communications, and all tasks from 0 to 20 incurred 10 to 12 min with both failure and security delays and impact on total delay. The 0 to 30 tasks are executed on fog and cloud nodes and 16 and 18 min and determined total delays for all tasks. Therefore, all tasks have higher delays if we manage them efficiently on different computing nodes.

We implemented the proposed scheme along with baseline approaches in the experiment. For instance, SFTS, IoT Without Federated, IoT With Federated, and IoT Fog Cloud schemes evaluate the performance of healthcare tasks on heterogeneous nodes. We evaluated the performance of all tasks in terms of total delays as determined in objective function *Total* as shown in Eq. (6). In the simulation result, as shown in Figure Fig. 10, the y-axis indicates that the different number of tasks determines *Total* time in minutes for all tasks. The SFTS scheme executed all tasks on different computing nodes with the *Total* delays in 2 min. The IoT With Federated scheme performed all tasks on different computing nodes with the *Total* delays in 4 min. The IoT Fog Cloud With Federated scheme executed all tasks on different computing nodes with the *Total* delays in 8 min. However, without federated learning, it has 10 min on different computing nodes for all tasks. We analyzed the performance of different architectures and schemes evaluated based on various tasks during the experiment. We analyzed and monitored the performance of all methods: SFTS, IoT Without Federated, IoT With Federated, and IoT Fog Cloud, and noted the *Total* total delays of all tasks in the environment. Fig. 10 shows the performance of different schemes, but SFTS outperformed all existing methods.

In our simulation criteria, *Total* variable for all tasks must be less given the deadlines of all tasks during execution on heterogeneous nodes. We determined the local processing delay during mobile offloading with the proposed scheme SFTS. The y-axis in Fig. 11 shows that all the mobile offloading tasks with the collected data take 5 to 7 min from sensor collection to mobile processing and during offloading in SFTS. The security and failure delay is also controlled under the given deadlines. We determined the total delays as *Total* incurred within 10 min during mobile offloading with 20 tasks. The baseline IoT With a Federated scheme executed all mobile processing tasks, including failure and security, within

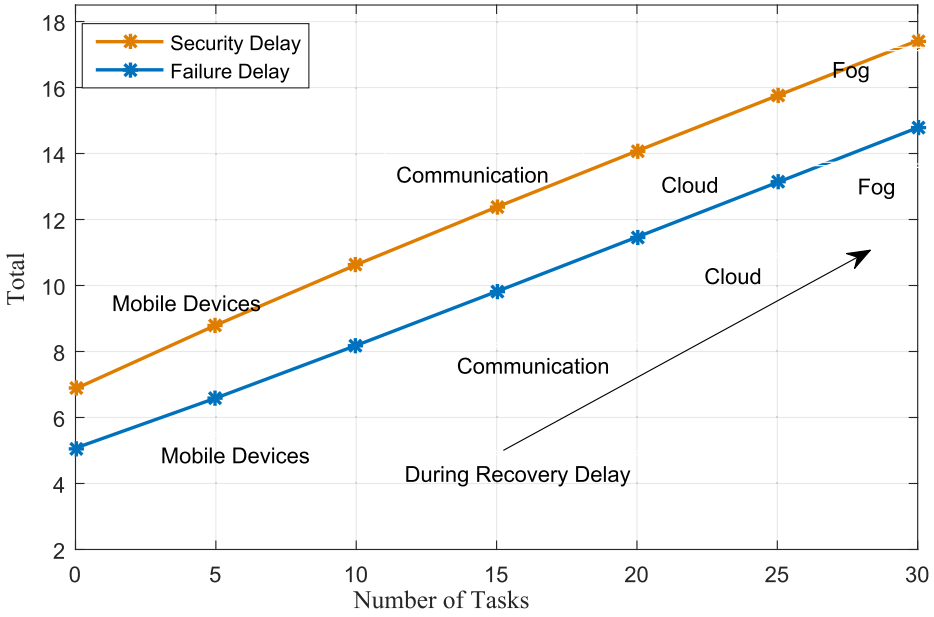

Fig. 9. Recovery total delays of tasks during security and failure issues in different computing nodes.

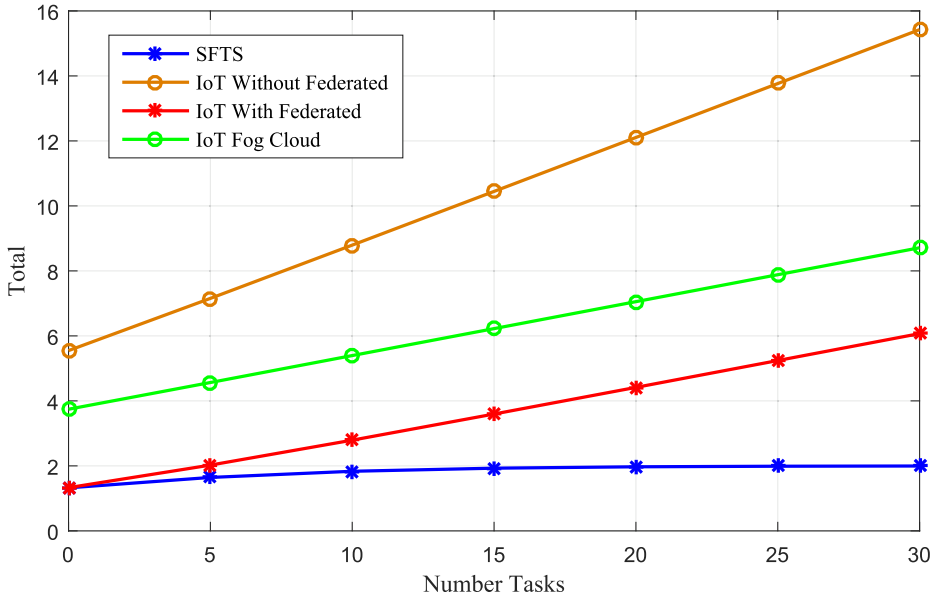

Fig. 10. Offloading and scheduling performance of healthcare tasks with different schemes on different computing nodes.

9 min with a random number of 20 tasks. Furthermore, the IoT Fog Cloud scheme incurred 9 min for mobile offloading, 10 min for communication, and 12 min during fog and cloud scheduling for all tasks. We analyzed the performance of different architectures, and schemes were evaluated based on various tasks during the experiment. We researched and monitored the performance of all strategies: SFTS, IoT Without Federated, IoT With Federated, and IoT Fog Cloud, and noted the *Total* total delays of all tasks in the environment. Fig. 11 shows the performance of different schemes, but SFTS outperformed all existing methods.

Our simulation considers the trade-off constraints between the resource consumption of different computing nodes and objective function *Total* for all tasks. The trade-off is always conflicting because there is less delay in the aim function *Total*, which consumes a higher ratio of computing resources. Therefore, we schedule all tasks based on given deadlines to handle the resource constraint issues of mobile devices and less resource consumption of fog and cloud nodes. These constraints, such as security, failure, and deadline, consume the resources of all computing nodes. We determined the failure and security-enabled scheduling with the checkpointing mechanism, where all nodes can resume failure of tasks from the point of failure. We analyzed the performance of different architectures and schemes evaluated based on diverse tasks during the experiment. We examined and monitored the performance of all strategies: SFTS, IoT Without Federated, IoT With Federated, and IoT Fog Cloud. We noted the *Total* total delays of all tasks in the environment. Fig. 12 shows the performance of different schemes, but SFTS outperformed all existing methods. Fig. 12

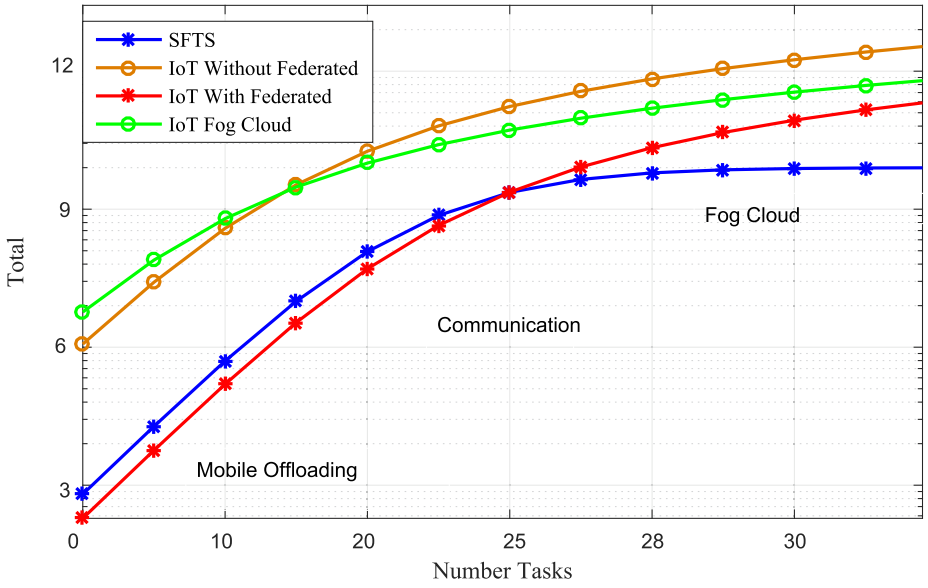

Fig. 11. Secure and failure enabled performances of schemes.

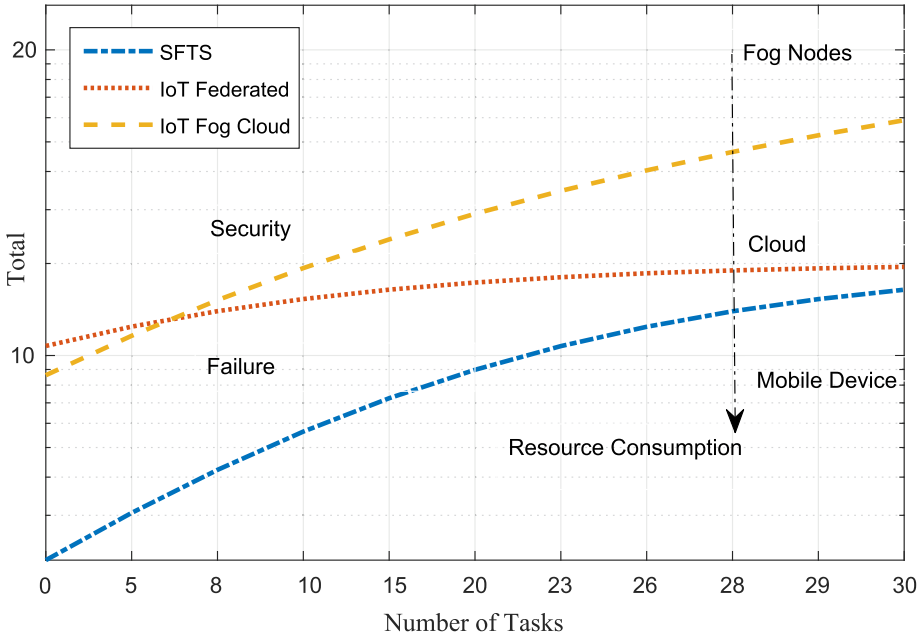

Fig. 12. Failure and security recovery performances of tasks with different schemes.

shows the proposed scheme SFTS executed all tasks with the minimum resource consumption under deadlines with 10-min delays. These approaches, IoT Federated and IoT Fog Cloud, consume many resources and are incurred with a total of 12 to 14 min of delays. The baseline approaches also consume much more resources due to scheduling all tasks without deadlines on available nodes.

Resource consumption is the key challenge in the IIoT healthcare environment. There are different kinds of resource consumption for the IIoT healthcare environment. The risk of security and resource availability failure is the key challenge for IIoT architectures. The offloading and scheduling are the schemes in which we maintained the resource consumption in the network. We analyzed the performance of different architectures and schemes based on different tasks during an experiment. We analyzed and monitored the performance of all schemes SFTS, IoT Without Federated, IoT With Federated, and IoT Fog Cloud and noted the *Total* total delays of all tasks in the environment. Fig. 13 shows the performance of different schemes, but SFTS outperformed all existing schemes. The different aspects are evaluated, such as failure ratio, security validation and resource leakage, and availability of fog and cloud nodes during offloading for processing. It has been observed that the baseline schemes with different IoT healthcare tasks have a failure ratio higher than SFTS. The security validation is more appropriate with the SFTS than baseline schemes.

We validated the work in different aspects, such as local processing delay, offloading communication delay (DT), fog processing

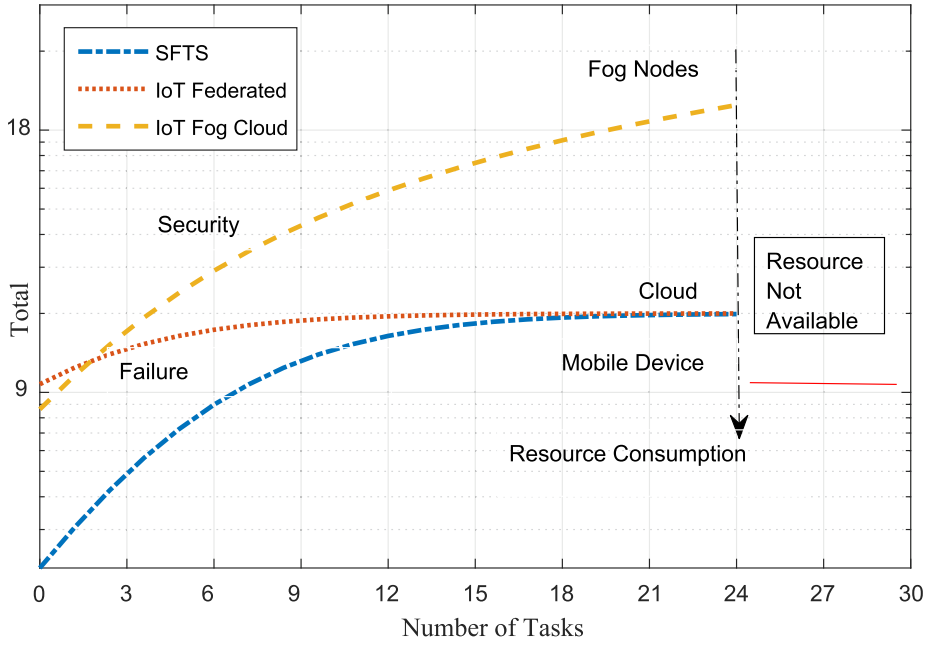

Fig. 13. Resource consumption of IIoHT sensory tasks with different failure and security schemes.

delay for training and validation, and cloud delay. We analyzed the failure and security aspects with the differences as shown in Table 7. The different scenarios such as Method, $T, L_t^e, C_t^e, Cloud_t^e, F_t^e, Total$, Failure, and Security are the constraints where performance evaluation can be compared and analyzed with the different methods. We can analyze the performance of the proposed scheme with different methods. For instance, SFTS obtained the results of tasks in different scenarios and different nodes, tasks t = 1–10, local processing = 6, offloading time = 8, fog delay = 6, cloud delay = 10, and total delay = 30, where only three tasks 3 are failed, 10 shows that, all tasks are successfully met the security between nodes without any failure of tasks. We analyzed the all results of all methods with different tasks from t = 1 to 30 with different methods as shown in Table 7.

Security validation in different nodes is a crucial process in different healthcare environments. Table 8 shows the data's encryption and decryption validation on different nodes. The parameters are considered in this phase, as shown in the following way. For instance, Task, Data, Encryption, Decryption, Validation, and Status are the encryption and decryption validations at the federated learning-enabled DT mobile fog cloud network. For instance, a task t = 1 has data 99, has encryption eO2aTgb6g2VL +83foW2jCw==, and again decrypts at 99. At the same time, these validations exist between $m1 \sim dt1$ with the status yes. Furthermore, this data is offloaded from the DT fog node to cloud computing. For instance, a task t = 1 with data 120, encryption KWuVYMKzEW0oPeb4ydQd7w==, and decryption 120, between $dt1 \sim s1$ with status yes. In this way, each piece of data from mobile devices to fog nodes and fog nodes to cloud computing is validated based on encryption and decryption with the shared keys federated by mobile fog cloud networks.

We discussed the different metrics that were identified during the simulation. We discussed these metrics, such as resource consumption, Scalability, and memory% during the execution of tasks in the simulator as shown in Table 9. We analyzed these metrics with the different baseline approaches and proposed methods for the different number of tasks, e.g., 30. We evaluated these metrics on the mobile node, fog nodes, and cloud computing. In our simulation, we keep the scalability fixed, which means in our current problem, we are not considering the resource provisioning with the cost constraints. Therefore, we keep the scalability fixed without scaling up and scaling down during the execution of tasks in the simulator. We determine the resource consumption in megabytes (MB) as shown in Table 9 for all methods. We determined the memory usage ratio for the execution of all tasks on different nodes with the different methods, as shown in Table 9. We can

Table 7
Work validation in different aspects and scenarios with different.

Method	T	L_t^e	C_t^e	$Cloud_t^e$	F_t^e	*Total*	Failure	Security
SFTS	t = 1–10	6	8	6	10	30	3	10
Without Federated	t = 1–10	16	18	16	10	60	3	10
IoT With Federated	t = 1–10	12	10	7	13	42	5	7
IoT Fog Cloud	30	t = 1–10	15	15	18	48	7	9
SFTS	t = 11–20	7	6	8	10	31	2	10
Without Federated	t = 11–20	17	17	17	10	61	3	8
IoT With Federated	t = 11–20	10	10	10	16	46	3	9
IoT Fog Cloud	30	t = 11–20	20	20	20	60	6	9
SFTS	t = 21–30	6	8	6	11	31	2	10
Without Federated	t = 21–30	16	20	16	10	62	4	8
IoT With Federated	t = 21–30	12	12	7	13	44	1	5
IoT Fog Cloud	t = 21–30	30	15	15	20	60	1	6

Table 8
Encryption and decryption detection schemes.

Task	Data	Encryption	Decryption	Validation	Status
t = 1	99	eO2aTgb6g2VL + 83foW2jCw==	99	$m1 \sim dt1$	yes
t = 1	120	KWuVYMKzEW0oPeb4ydQd7w==	120	$dt1 \sim s1$	yes
t = 2	155	K7 + 93/Wi0ecCKP14ySmoeg==	155	$m1 \sim dt2$	yes
t = 2	1000	fQrhvbmiJto19Y9/JAyRiQ==	1000	$dt2 \sim s2$	yes

Table 9
Encryption and decryption detection schemes.

Method	Task	Node	Resource Consumption	Scalability	Memory%
SFTS	30	Mobile	380 MB	Fixed	0.3
SFTS	30	Fog Nodes	1000 MB	Fixed	0.7
SFTS	30	Cloud	1500 MB	Fixed	0.9
IoT Without Federated	30	Mobile	500 MB	Fixed	0.8
IoT Without Federated	30	Fog Nodes	1500 MB	Fixed	0.9
IoT Without Federated	30	Cloud	2000 MB	Fixed	0.12
IoT With Federated	30	Mobile	500 MB	Fixed	0.7
IoT With Federated	30	Fog Nodes	1600 MB	Fixed	0.14
IoT With Federated	30	Cloud	2200 MB	Fixed	0.16
IoT Fog Cloud	30	Mobile	700 MB	Fixed	0.9
IoT Fog Cloud	30	Fog Nodes	2000 MB	Fixed	0.19
IoT Fog Cloud	30	Cloud	3000 MB	Fixed	0.21

observe from Table 9 that SFTS consumes less resources and memory than existing methods on mobile fog and cloud nodes during the execution of tasks. The main reason is that we implemented the replica of the data processing based on digital twins. Therefore, the results and data recovery from the cloud are migrated or downloaded to the fog nodes for scheduling. It is a robust and efficient way to use the digital twin-enabled fog and cloud nodes for the distributed Iot applications and minimize the resource consumption and memory usage for similar tasks during security and failure situations in our architecture.

5.3. Findings and shortcomings of SFTS and baseline algorithms

In this study, we presented the SFTS method, which consists of different schemes such as local processing, federated learning-enabled fog and cloud nodes, and digital twin. The main finding of the SFTS is to execute all IoT content tasks with minimum delays. The total delays as we determined in Eq. (6) combinations of different delays. Therefore, we scheduled all IoT tasks to the different computing nodes with minimum delays. In the result discussion, we showed the findings and limitations of SFTS for all tasks on mobile, fog, and cloud nodes. We have implemented the four baseline strategies, IoT Without Federated learning, IoT With Federated learning, and IoT Fog Cloud schemes for IoT content tasks in the simulation environment. We considered the different constraints such as security, processing delay, deadline, resource consumption, and failure of tasks. We analyzed the performances of all algorithms as shown in the result analysis and discussion with the IoT random number of tasks in fog and cloud networks. We investigated IoT tasks' security and failure delay constraints on mobile, fog, and cloud nodes with all algorithms designed during execution in the architecture. This baseline IoT Without a Federated learning strategy, scheduled all mobile, fog, and cloud network tasks. The strategy IoT Without Federated learning scheduled all tasks on different computing nodes in a secure and delay-efficient form. However, as all simulation results show, this strategy has suffered higher delays. The main reason is that, in this strategy, the main node made all the decisions during the security recovery and failure of tasks and acknowledged all nodes. That means all nodes are clustered in the network, but decisions made by centralized nodes suffer higher delays for all tasks. In the previous higher delays, all the tasks missed their deadlines and degraded the performances of applications. This IoT Fog Cloud strategy divided the scheduling tasks decision among different computing nodes based on the scheduler and got less delay than IoT without the federated strategy. The main reason is that all the schedulers can communicate with each other and reschedule the failure of tasks from the point of failure. However, due to many constraints, such as local processing, communication, failure, security, and remote processing, this strategy suffered from delays analyzed one constraint at a time. The federated learning divided the computing analyzing of different at different fog and cloud and aggregated to the centralized nodes. However, one node's failure in federated learning still suffers from higher delays in the network. SFTS is the optimal strategy that integrates the digital twin on fog nodes, where cloud nodes are replicated and executed on fog nodes. The digital twin is adaptively integrated with the cloud nodes, where, based on federated learning, we can exchange their updates to all connected nodes. Therefore, the SFTS strategy for IoT tasks in the network can easily manage all the constraints on different computing nodes. However, there are still limitations in the SFTS scheme for broad-level infrastructure. All the proposed and baseline strategies did not consider the wait time of tasks before scheduling in the mobile fog cloud networks. The baseline strategies and proposed work widely miss task power consumption, cost, and sustainability. These limitations still need to improve the total delays and time complexity to become efficient for all time-sensitive tasks with the given priority. Therefore, all baseline and proposed approaches must address these limitations in future works. In the current version of methods, we did not consider resource provisioning to avoid resource scalability. In this work, we only exploited the fixed type of resources. We keep the fixed scalability in the current architecture version and do not consider resource provisioning. However, the diversity features can increase the ratio of tasks and types. Therefore, power consumption on local devices and wireless networks could be increased. In future work, we consider the power consumption, cost, resource provisioning, scalability, and existing constraints with the more robust and adaptive schemes.

6. Conclusion

Based on performance evaluation, the study analyzed and processed the different healthcare sensory data to monitor and predict

the healthcare data in distributed mobile fog and cloud networks. This paper presented the secure, fault-tolerant, empowered wearable healthcare sensors aware Industrial Internet of Things (IIoT) Framework based on digital twin federated fog-cloud models. The study aims to reduce the time and resources needed to process healthcare sensor data for security, task execution, and fault tolerance. The study presents the Secure and Fault-Tolerant Scheme (SFTS) algorithm framework that optimizes the IoT sensor data and executes the healthcare data with the minimum offloading and processing delays. Simulation results show that the proposed work minimized the security risk by 40%, failure risk of tasks risk by 50%, and the training and testing time by 39% for all IIoT healthcare tasks in mobile fog cloud networks.

The study will implement blockchain technology in future work and consider the different healthcare clinics for sensory data in heterogeneous computing nodes. The current work version has power consumption, electricity cost, and distributed resource sharing that have yet to be considered in the present work. In future work, we will add more constraints in the considered framework with the discrete and polynomial time for all tasks. There exist constraints in the SFTS that relate to the time complexity associated with diverse limitations. To illustrate, mobile device waiting time during offloading has yet to be established. As a result, tasks within the scope of IoT might encounter challenges with meeting deadlines and maintaining performance due to prolonged waiting intervals. The difficulty of achieving load balance in federated learning between fog and cloud nodes endures. Consequently, these considerations tied to the equilibrium of fog and cloud nodes lead to elongated waiting duration. This circumstance significantly impacts the time complexity of IoT fog cloud tasks within the framework of our architecture. Therefore, in future work, we will consider the delays for IoT fog cloud tasks in our extended architecture.

Author Contributions

All authors contributed equally to the final dissemination of the research investigation as a full article. All authors have read and agreed to the published version of the manuscript.

Ethical Approval

The manuscript does not report on or involve the use of any animal or tissue and is “Not applicable” to this manuscript.

Funding

This article was co-funded by the European Union under the REFRESH - Research Excellence For REgion Sustainability and High-tech Industries project number CZ.10.03.01/00/22_003/0000048 via the Operational Programme Just Transition. Also, this work was supported by the Ministry of Education, Youth and Sports of the Chezk Republic conducted by VSB - Technical University of Ostrava, Czechia under Grants SP2023/039 and SP2023/042.

Data Availability Statements

The study exploited the public healthcare sensors, and datasets are publically available on the following URL. https://github.com/ABDULLAH-RAZA/IIoThealthcare-Sensors-Data. The datasets have different features such as ECG Lead-1, Lead-2, Ankle, BP (blood pressure), Temp (Temperature), HB (Heartbeats), CH (Cholesterol) speed, location, real-time, stayed Activity, and Users.

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.

References

Akash, S.S., Ferdous, M.S., 2022. A blockchain based system for healthcare digital twin. IEEE Access 10, 50523–50547.
Alaasam, A.B., Radchenko, G., Tchernykh, A., González Compeán, J., 2020. Analytic study of containerizing stateful stream processing as microservice to support digital twins in fog computing. Programm. Comput. Softw. 46, 511–525.
Alshathri, S., Hemdan, E.E.-D., El-Shafai, W., Sayed, A., 2023. Digital twin-based automated fault diagnosis in industrial iot applications. CMC-Comput. Mater. Continua 75 (1), 183–196.
Azzaoui, A.E., Kim, T.W., Loia, V., Park, J.H., 2021. Blockchain-based secure digital twin framework for smart healthy city. Adv. Multimedia Ubiquit. Eng. 716, 107.
Chi, H.R., Wu, C.K., Huang, N.-F., Tsang, K.F., Radwan, A., 2022. A survey of network automation for industrial internet-of-things towards industry 5.0. IEEE Trans. Industr. Inf.
Daigneault, J., St-Hilaire, M., 2021. Profit maximization model for the task assignment problem in 2-tier fog/cloud network environments. IEEE Network. Lett. 3 (1), 19–22.
Darvishi, H., Ciuonzo, D., Rossi, P.S., 2021. Real-time sensor fault detection, isolation and accommodation for industrial digital twins. In: 2021 IEEE International Conference on Networking, Sensing and Control (ICNSC), vol. 1. IEEE, pp. 1–6.
Dharangan, B., Praveen, J., Rajagopal, S., Jegajothi, B. et al., 2022. Secure cloud-based e-health system using advanced encryption standard. In: 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE, pp. 642–646.
Elayan, H., Aloqaily, M., Guizani, M., 2021. Digital twin for intelligent context-aware iot healthcare systems. IEEE Internet Things J. 8 (23), 16 749–16 757.
el Azzaoui, A., Kim, T.W., Loia, V., Park, J.H., 2020. Blockchain-based secure digital twin framework for smart healthy city. In: Advanced Multimedia and Ubiquitous Engineering: MUE-FutureTech 2020. Springer, pp. 107–113.
Ghita, M., Siham, B., Hicham, M., Abdelhafid, A.E.M., Laurent, D., 2020. Geospatial business intelligence and cloud services for context aware digital twins development. In: 2020 IEEE International conference of Moroccan Geomatics (Morgeo). IEEE, pp. 1–6.
Haleem, A., Javaid, M., Singh, R.P., Suman, R., 2023. Exploring the revolution in healthcare systems through the applications of digital twin technology. Biomed. Technol. 4, 28–38.
Jean Effil, N., Rajeswari, R., 2022. Wavelet scattering transform and long short-term memory network-based noninvasive blood pressure estimation from photoplethysmograph signals. SIViP 16 (1), 1–9.
Jimenez, J.I., Jahankhani, H., Kendzierskyj, S., 2020. Health care in the cyberspace: Medical cyber-physical system and digital twin challenges. Digital Twin Technol. Smart Cities, 79–92.
Khan, S., Arslan, T., Ratnarajah, T., 2022. Digital twin perspective of fourth industrial and healthcare revolution. IEEE Access 10, 25732–25754.
Khoso, F.H., Lakhan, A., Arain, A.A., Soomro, M.A., Nizamani, S.Z., Kanwar, K., 2021. A microservice-based system for industrial internet of things in fog-cloud assisted network. Eng. Technol. Appl. Sci. Res. 11 (2), 7029–7032.
Kim, S., 2020. New application task offloading algorithms for edge, fog, and cloud computing paradigms. Wireless Commun. Mobile Comput. 2020, 1–14.
Konigsburg, J.A., 2022. Modern warfare, spiritual health, and the role of artificial intelligence. Religions 13 (4), 343.
Lakhan, A., Mohammed, M.A., Kadry, S., Abdulkareem, K.H., Al-Dhief, F.T., Hsu, C.-H., 2021. Federated learning enables intelligent reflecting surface in fog-cloud enabled cellular network. PeerJ Comput. Sci. 7, e758.
Lakhan, A., Mohammed, M.A., Abdulkareem, K.H., Jaber, M.M., Nedoma, J., Martinek, R., Zmij, P., 2022. Delay optimal schemes for internet of things applications in heterogeneous edge cloud computing networks. Sensors 22 (16), 5937.
Leng, J., Sha, W., Wang, B., Zheng, P., Zhuang, C., Liu, Q., Wuest, T., Mourtzis, D., Wang, L., 2022. Industry 5.0: Prospect and retrospect. J. Manuf. Syst. 65, 279–295.
Leong, Y.K., Tan, J.H., Chew, K.W., Show, P.L., 2021. Significance of industry 5.0. In: The Prospect of Industry 5.0 in Biomanufacturing. CRC Press, pp. 95–114.
Lv, Z., 2023. Digital twins in industry 5.0. Research 6, 0071.
Naha, R.K., Garg, S., Chan, A., Battula, S.K., 2020. Deadline-based dynamic resource allocation and provisioning algorithms in fog-cloud environment. Future Generat. Comput. Syst. 104, 131–141.
Nguyen, T.N., Ponciroli, R., Bruck, P., Esselman, T.C., Rigatti, J.A., Vilim, R.B., 2022. A digital twin approach to system-level fault detection and diagnosis for improved equipment health monitoring. Ann. Nucl. Energy 170, 109002.
Panda, S.K., Nanda, S.S., Bhoi, S.K., 2022. A pair-based task scheduling algorithm for cloud computing environment. J. King Saud Univ.-Comput. Infor. Sci. 34 (1), 1434–1445.
Rashid, L., Rubab, S., Alhaisoni, M., Alqahtani, A., Alsubai, S., Binbusayyis, A., Bukhari, S.A.C., 2022. Analysis of dimensionality reduction techniques on internet of

things data using machine learning. Sustain. Energy Technol. Assessm. 52, 102304.
Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H.R., Albarqouni, S., Bakas, S., Galtier, M.N., Landman, B.A., Maier-Hein, K., et al., 2020. The future of digital health with federated learning. NPJ Digital Med. 3 (1), 119.
Salman, A.O., Geman, O., 2023. Evaluating three machine learning classification methods for effective covid-19 diagnosis. Int. J. Mathe. Stat. Comput. Sci. 1, 1–14. https://doi.org/10.59543/ijmscs.v1i.7693.
Unnisa, S., Vijayalakshmi, A., Jagun, Z.T., 2023. Deep neural network architecture and applications in healthcare. Deep Learn. Healthcare Decis. Mak., 25
Volkov, I., Radchenko, G., Tchernykh, A., 2021. Digital twins, internet of things and mobile medicine: a review of current platforms to support smart healthcare. Programm. Comput. Softw. 47, 578–590.
Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., Wang, F., 2021. Federated learning for healthcare informatics. J. Healthcare Informat. Res. 5, 1–19.
Yang, Z., Yang, Y., Xu, C., 2022. Demonstration on unblocking checkpoint for fault-tolerance in pregel-like systems. In: Web and Big Data: 6th International Joint Conference, APWeb-WAIM 2022, Nanjing, China, November 25–27, 2022, Proceedings, Part III. Springer, pp. 456–460.
Younan, M., Houssein, E.H., Elhoseny, M., Ali, A.A., 2020. Challenges and recommended technologies for the industrial internet of things: A comprehensive review. Measurement 151, 107198.
Zhang, J., Li, L., Lin, G., Fang, D., Tai, Y., Huang, J., 2020. Cyber resilience in healthcare digital twin on lung cancer. IEEE Access 8, 201900–201913.