Received: 14 June 2024 | Revised: 18 November 2024 | Accepted: 21 November 2024

DOI: 10.1049/wss2.12101

IET Wireless Sensor Systems

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Implementation and evaluation of digital twin framework for Internet of Things based healthcare systems

Ahmed K. Jameil[1,2] | **Hamed Al-Raweshidy**[1]

[1]College of Engineering, Design and Physical Sciences, Brunel University London, London, UK

[2]Department of Computer Engineering, University of Diyala, Baqubah, Iraq

Correspondence

Hamed Al-Raweshidy and Ahmed K Jameil.
Email: hamed.al-raweshidy@brunel.ac.uk and 2006957@brunel.ac.uk

Funding information

Brunel University of London; Lloyd's Register

Abstract

The integration of digital twins (DTs) in healthcare is critical but remains limited in real-time patient monitoring due to challenges in achieving low-latency telemetry transmission and efficient resource management. This paper addresses these limitations by presenting a novel cloud-based DT framework that optimises real-time healthcare monitoring, providing a timely solution for critical healthcare needs. The framework incorporates a Pyomo-based dynamic optimisation model, which reduces telemetry latency by 32% and improves response time by 52%, surpassing existing systems. Leveraging low-cost, low-latency multimodal sensors, the system continuously monitors critical physiological parameters, including SpO2, heart rate, and body temperature, enabling proactive health interventions. A DT definition language (Digital Twin Definition Language)-based time series analysis and twin graph platform further enhance sensor connectivity and scalability. Additionally, the integration of machine learning (ML) strengthens predictive accuracy, achieving 98% real-time accuracy and 99.58% under cross-validation (cv = 20) using the XGBoost algorithm. Empirical results demonstrate substantial improvements in processing time, system stability, and learning capacity, with real-time predictions completed in 17 ms. This framework represents a significant advancement in healthcare monitoring, offering a responsive and scalable solution to latency and resource constraints in real-time applications. Future research could explore incorporating additional sensors and advanced ML models to further expand its impact in healthcare applications.

KEYWORDS

cloud computing, patient monitoring, real-time systems, sensors, telemedicine

1 | INTRODUCTION

A digital twin (DT), commonly abbreviated as DT, is referred to as a virtual model that mirrors a tangible entity and is characterised by dynamic, reciprocal connections. Digital twin is defined as a virtual representation of a physical entity that facilitates real-time data exchange between its virtual and physical counterparts. In healthcare, DTs are presented as a breakthrough solution, addressing challenges, for example, early detection of medical conditions or monitoring of chronic illnesses in real time. Monitoring a heart patient's DT allows healthcare providers to, for example, assess the probability of an impending cardiac event, thereby enabling timely interventions. In the future, the use of DTs is anticipated to significantly influence personalised therapies and interventions [1, 2].

Advanced optimisation techniques such as Pyomo have been integrated into healthcare systems to optimise decision-making processes in DTs. For instance, Pyomo has been applied to optimise complex systems such as healthcare logistics and patient monitoring, demonstrating its versatility in resource allocation and real-time data management [3, 4]. These capabilities align well with the dynamic and evolving nature of DTs, allowing for enhanced efficiency and performance in healthcare applications.

Digital technologies and services have been demonstrated to be beneficial for healthcare professionals and patients alike, as they facilitate data collection, clinical communication, disease management, and other related functions [5–7]. Furthermore, DT has the potential to bridge gaps in current healthcare systems, that is, the delay in getting patient data during emergencies or in remote regions, offering the potential benefits of immediate diagnostic insights and timely medical interventions [8].

The objective of Digital Twin for Healthcare (DTH) is to digitally replicate select aspects of human anatomy and physiology, specifically the organs of the human body, through the creation of a digital lifestyle. Although there have been some developments in the field of cardiology, the utilisation of DTH remains in its nascent stage, and its extensive adoption for the betterment of public health is expected to take several years [9]. In contrast, the monitoring, optimisation, and planning capabilities of DTs render them efficacious in enhancing population health as constituents of public health management systems [10].

Furthermore, advanced guidance on the implementation of a digital transformation is provided by academia, making it significantly difficult to identify a universally applicable solution for its execution. Digital twin has been employed in academic and professional contexts to investigate the impact of social distancing, as evidenced by studies conducted by Punn et al. [11]. The inaccessibility of implementation architecture information is a recurring issue in research pertaining to Digital Transformation. Despite the limited number of researchers who have directed their efforts towards the key components required for the production of a DT, a reference design has not yet been established, as evidenced by the literature [12].

Building on the challenges and limitations described previously, our study proposes an architecture that integrates cloud computing, IoT, machine learning (ML), and artificial intelligence (AI) to remotely monitor and assess patient health. Through sensors, data is transmitted to the cloud. A virtual patient replica, utilising this framework, offers monitoring, trend prediction from medical history, and collaborative patient scenarios. The primary contributions of this study are as follows:

1. The proposal of a novel DT architecture, based on the cloud and healthcare wearables, integrated with a Pyomo-based dynamic optimisation model. This architecture serves as a foundation for a DTH scenario and addresses the challenges of real-time monitoring, improving system scalability, and resource management, while enhancing the accuracy of emergency alerts for patients.
2. The presentation of a study on DTH using ML for comparison, diagnosis, and prediction, ensuring consistent results by comparing seven different ML algorithms.
3. The proposal of a cost-effective DT simulation framework for twin graphs, using JavaScript Object Notation for Linked Data (JSON-LD) and sensors, for monitoring and health tracking in humans, utilising pay-as-you-go cloud services.
4. This study also aims to validate the design through a comparison of physical and digital data utilising time series insight (TSI), with Flask used to validate the ML model. Latency calculations were conducted and the results indicate a relatively low value compared to prior studies.

The remaining sections of this study are structured as follows: In Section 2, the background and foundational concepts of DT technology are discussed. In Section 3, a review of the available literature on DT in healthcare is provided. A cloud-based DT architecture is proposed in Section 4. Section 5 describes the framework developed on the basis of the twin graph. The implementation setup is presented in Section 6. Section 7 describes Proof of Concept (PoC). The Results and Discussions are presented in Section 8. Finally, Section 9 provides the conclusion and future work for this study.

2 | BACKGROUND

The digital transformation process that is currently transforming many industries, including health, began with the launch of the Industry 4.0 project in 2013. The approach relies heavily on advanced technologies, for example, IoT, cloud and edge computing, AI, and big data analytics [13–15]. The DT paradigm, based on the aforementioned technologies, allows for the digital transformation of any system and is commonly utilised by industrial and engineering companies. Over the previous decade, DT technology has been recommended for healthcare applications. One of the most impressive DT applications is the Healthcare DT (DTH) [16].

In the realm of Industry 5.0, emphasis is placed on addressing human needs, from healthcare provision to the fulfilment of personal growth and self-actualisation goals. This shift has catalysed a transformation within the healthcare sector, where DT technology has been adopted to embed human-centric approaches in intelligent manufacturing systems, thereby enhancing rapid diagnostics and monitoring [17]. Nevertheless, the ambition to synchronise the physical and digital dimensions through DT encounters notable hurdles, chiefly the challenge of achieving instantaneous alignment between the two spheres [18, 19].

In the research on DTs, optimisation techniques were pivotal in managing complex, resource-driven healthcare systems. The Pyomo optimisation library was utilised in various fields, demonstrating its versatility in solving resource allocation and logistical challenges [3]. Though not always directly applied to healthcare, Pyomo's capacity to optimise complex systems was highlighted. Its combination with solvers like CPLEX proved effective for tasks such as healthcare logistics, energy systems, and predictive modelling [20, 21]. This demonstrated Pyomo's potential for real-time healthcare monitoring and resource management, improving efficiency and reducing latency.

The research in DT further encompasses the enhancement of wireless body area networks, alongside the deployment of advanced signal processing and sensors that support DT, with the integration of Markov decision processes and AI to elevate both efficiency and reliability in health monitoring systems

[22–24]. Within smart homes, DT applications strive to improve the monitoring, prediction, and control of health parameters, utilising an array of wireless and wearable technologies [25, 26]. In the healthcare sector, professionals utilise DT in conjunction with cloud and IoT-edge computing, blockchain, and ML, aiming to deliver intelligent predictive diagnostics and secure health data management. This DT framework significantly advances monitoring capabilities [27, 28].

Additionally, DT has been employed to devise frameworks that aid in clinical monitoring, evaluating patient needs, and identifying emergency risks promptly [29]. For instance, in thoracic healthcare, the Lung-DT framework integrates AI with historical radiological data and IoT sensor inputs to accurately classify lung diseases, thereby improving upon traditional diagnostic methodologies. However, challenges, that is, the absence of data storage within this framework pose hurdles for end-users [30]. The concept of the Virtual Human Twin has been introduced, offering a detailed digital representation of human pathophysiology and suggesting a cooperative infrastructure, which is instrumental in advancing the development and uptake of DT in healthcare [31, 32].

3 | RELATED WORK

Grieves developed a comprehensive framework for the DT model, introducing a foundational concept that models a system in three dimensions: physical object, virtual entity, and link. This framework laid the groundwork for DT applications across various domains, demonstrating the potential for accurate, interconnected system representations [33]. Building on this, Tao et al. extended the model to a five-dimensional architecture, incorporating DT-related information and resources to enhance the DT framework's capacity for representing complex systems in greater detail [34].

In healthcare, researchers have applied DT models by integrating big data and AI-driven models to simulate human physiology, showing potential to provide tailored clinical solutions. However, despite this promise, significant technological, privacy, and ethical challenges still limit the practical deployment of DTs in healthcare [35].

A novel DT approach was investigated by Yang et al. in which cardiovascular casts, CT scans, and simulation algorithms were used to operate ultrasound probes in virtual environments. This approach contributed to the development of a method for learning standard views and anatomical relationships in virtual foetal heart imaging, though limitations were observed due to inconsistencies with real foetal heart structures, restricting its practical application in healthcare [36].

A framework focused on high-fidelity cardiac electrophysiology DTs using clinical 12-lead electrocardiogram (ECGs) was developed by Gillette et al., and challenges in creating precise anatomical and functional twins for cardiac care were addressed. Although this framework represents an advancement in accurate cardiac DT modelling, the integration of cloud computing and IoT for real-time monitoring was not included. As a result, essential components, such as real-time data handling, visualisation, analysis, and latency considerations, were omitted, with a primary focus placed on managing costs [37, 38].

In ref. [39], an automated gait data control system for a fully actuated lower limb exoskeleton DT was proposed for medical rehabilitation. Through this approach, improvements in interaction, autonomy, and safety were achieved via simulation trials. The application of deep learning and big data analytics within healthcare DTs was explored by Lv et al., where sensor data was analysed for real-time health monitoring, representing a significant advancement in digital health solutions. However, considerations for cost and latency were not incorporated into their framework [40].

A cloud-based framework for elderly healthcare, named CloudDTH, was developed by Liu et al., combining big data, cloud computing, and IoT to deliver accurate and efficient healthcare services using DT technology. Despite these advancements, the framework does not thoroughly address real-time data latency issues or cost implications, nor are details provided on integrating ML for predictive analytics within healthcare DTs [41].

A decentralised architecture for the Industrial Internet of Things was developed by Lin et al., utilising blockchain, oracles, and DTs to facilitate secure and efficient data exchanges and computing collaborations between physical and digital entities. However, the integration of cloud computing and IoT for real-time health monitoring was not addressed, and discussions on latency issues and cost implications were also omitted [42]. Deep learning methodologies, such as Artificial Neural Networks, have been used to create a robust framework for analysing health datasets, ensuring precise and timely health insights [43]. Personal DT (PDT) technology was explored by Sahal et al., who demonstrated its potential to transform healthcare by enabling precise decision-making and personalised treatment choices [44].

Recent developments have highlighted the application of optimisation techniques within DT frameworks. For example, the use of Pyomo in optimising resource allocation and logistics in healthcare and other sectors has demonstrated significant flexibility in handling complex systems and enabling real-time decision-making [3, 20]. Through these applications, a foundation has been established for more efficient resource management and latency reduction in critical healthcare contexts, which is essential for advancing DT technologies in this domain.

In ref. [45], DT architecture and robotic systems were applied to enable reliable testing of synthetic soft tissue products, demonstrating potential for advancing medical device and biomechanics research. A DT-based, AI-driven analysis of disease parameters and patient data was proposed by Voigt et al., contributing to improved diagnostic accuracy, treatment outcomes, and preventative healthcare measures [46]. Research in ref. [47] focused on the development of medical cyber-physical platforms with multiple layers for data collection, data processing, cloud infrastructure, and actuation. In addition, wearable devices and AI were used by Chen et al.

to analyse data and simulate human processes, enhancing the understanding of user motivations, emotions, and preferences to improve user experience [48].

A smart clothing system with multiple sensors based on DT technology was developed by Yu et al. to address user interaction limitations in existing smart clothing systems. However, considerations for integrating cloud computing for real-time data processing, as well as the impact of latency and cost on system performance, were not explicitly included in this study [49]. Additionally, the use of actigraphs has been highlighted as crucial in applications such as interactive games, health monitoring, and bipolar illness prediction, contributing to improved well-being and allowing assessment of physical activity levels for psychological evaluation [50]. To achieve greater dependability, predictability, and accuracy in therapeutic results, computer simulations were proposed for collaboration with tissue engineering by the authors of [51].

In ref. [52], a framework for DT in remote surgery was presented, offering a foundational approach for applying DTs in surgical settings. A cardio-twin structure was introduced to diagnose ischaemic heart disease at the edge, providing an innovative solution for decentralised cardiac diagnosis [53]. Additionally, a convolutional neural network was employed to differentiate non-myocardial diseases from cardiac conditions using Physio Bank's ECG database, achieving an accuracy of 85.77% and a classification time of 4.8 s, contributing to more efficient cardiac disease differentiation [54].

Zhong et al. introduced Interactive Digital Twin Virtual Reality (IDTVR), a cloud-based framework designed to create interactive DT environments using virtual reality (VR) technology, demonstrating the potential of VR in DT applications for rehabilitation. However, latency issues and the cost implications of large-scale deployment were not extensively addressed, and the study focused primarily on rehabilitation without exploring broader healthcare monitoring applications [55]. Jia et al. proposed a concurrent end-to-end synchronisation and multi-attribute data resampling-enabled DT scheme, aiming to improve modelling accuracy and efficiency. Despite these advancements, the study does not extensively consider latency issues, cost implications, or broader healthcare monitoring capabilities [56].

Recent developments in DT systems for asthma home monitoring have enhanced reliability, efficacy, and capacity to evaluate breathing strategies, triggers, and environmental data [57]. Constant performance in patient data evaluation and interpretation has highlighted the dependability of DT systems in practical medical contexts. The management and treatment of asthma through DT-enabled mobile applications, gadgets, and remote monitoring systems have facilitated early intervention [58]. A wide range of DT applications in healthcare has indicated DT's potential to improve patient care and enable continuous monitoring. A comparative analysis of these frameworks and our proposed system, shown in Table 1, highlights key criteria, such as real-time monitoring, resource efficiency, and latency reduction, demonstrating how gaps identified in previous studies are addressed.

In summary, valuable insights into DT and healthcare applications have been provided by these studies, though gaps remain in addressing real-time data processing, resource efficiency, and latency reduction. In response, a solution is proposed through the integration of cloud computing, IoT, and the Pyomo optimisation model to enhance real-time monitoring, decision-making, and resource allocation. Higher accuracy, reduced latency, and greater scalability are achieved, offering a comprehensive and efficient healthcare monitoring solution.

4 | PROPOSED ARCHITECTURE OF DIGITAL TWIN

The proposed DT architecture was designed using cloud computing and IoT to enable scalability across healthcare facilities and patients. Platform-as-a-Service (PaaS) was utilised to ensure seamless integration of additional sensors and devices without affecting system performance, while edge computing was employed to reduce latency by offloading computational tasks to near-edge devices. Cloud platforms,

TABLE 1 Comparison of various frameworks including our work.

Ref.	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
[59] 2020	✓	✓	–	–	–	–	–	–	–	–	–
[60] 2021	✓	✓	–	✓	–	–	–	–	–	–	–
[61] 2023	✓	✓	✓	–	–	–	–	–	–	–	–
[62] 2023	✓	✓	–	–	–	–	–	–	✓	–	–
[63] 2023	✓	–	–	✓	–	–	–	✓	✓	–	–
[64] 2024	✓	✓	–	–	–	–	–	–	✓	–	–
[65] 2024	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	✓
This work	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note: Criteria: C; C1: Health; C2: IoT; C3: Temporal Analysis; C4: Adversity; C5: Predictive Decision; C6: Time-Specific; C7: Real-time; C8: Stability; C9: Security; C10: Resource Efficiency; C11: Latency Reduction.

Abbreviations: BS, base station; GLPK, GNU linear programming kit; SSL, secure sockets layer; TLS, transport layer security; UE, user equipment.

such as Azure, supported horizontal scaling, maintaining system robustness as patient numbers and data volumes increased. Figure 1 illustrates the architecture, integrating both cloud and edge computing for real-time analysis of mobile health data.

The cloud-based architecture consists of six layers, integrating cloud and edge computing to enable real-time data analysis and monitoring. Algorithm 1 outlines the monitoring and prediction process for patient health indicators. The system collects key health metrics through IoT devices, processes the data to evaluate fluctuations, and checks whether these fall within predefined thresholds.

In Section 6, we will explain in detail the context of ML processing, where the data undergoes an additional normalisation procedure, resulting in the updating of the DT model.

4.1 | Problem formulation

Implementing DTs in healthcare faces several critical challenges, especially in real-time monitoring. The main obstacles are as follows:

1. Telemetry Latency: High-latency transmission delays health data updates, hindering timely patient monitoring. Reducing latency is essential for prompt medical interventions.
2. Resource Management: Efficient allocation of resources across cloud and edge infrastructures is vital to ensure scalability and cost-efficiency without compromising performance.
3. Scalability and System Stability: As the scale of patient monitoring increases, maintaining both system stability and scalability becomes more complex, especially in managing the large volumes of data generated by multimodal sensors.
4. Real-Time Monitoring and Prediction: Continuous and accurate real-time monitoring of vital physiological parameters (e.g. SpO2, heart rate (HR), and body temperature (BT)) remains challenging, particularly when predictive analytics are required for proactive healthcare decisions.

To address these limitations, a cloud-based DT architecture is proposed in this paper. By incorporating a Pyomo-based dynamic optimisation model, multimodal sensor integration, and ML algorithms, the system is designed to reduce telemetry latency, enhance response time, and improve predictive accuracy in real-time healthcare monitoring. This approach enables scalable, low-latency, and resource-efficient monitoring across diverse environments. The following provides a detailed explanation of the layers in the proposed DT architecture, illustrating how each component contributes to addressing the identified challenges.

4.2 | Device layer

On the left side of Figure 1 is the device layer of this design architecture. This layer shows the controller, a NodeMCU ESP8266 controller, as well as the sensors that a person wears. The NodeMCU ESP8266 transmits data to the IoT Hub using

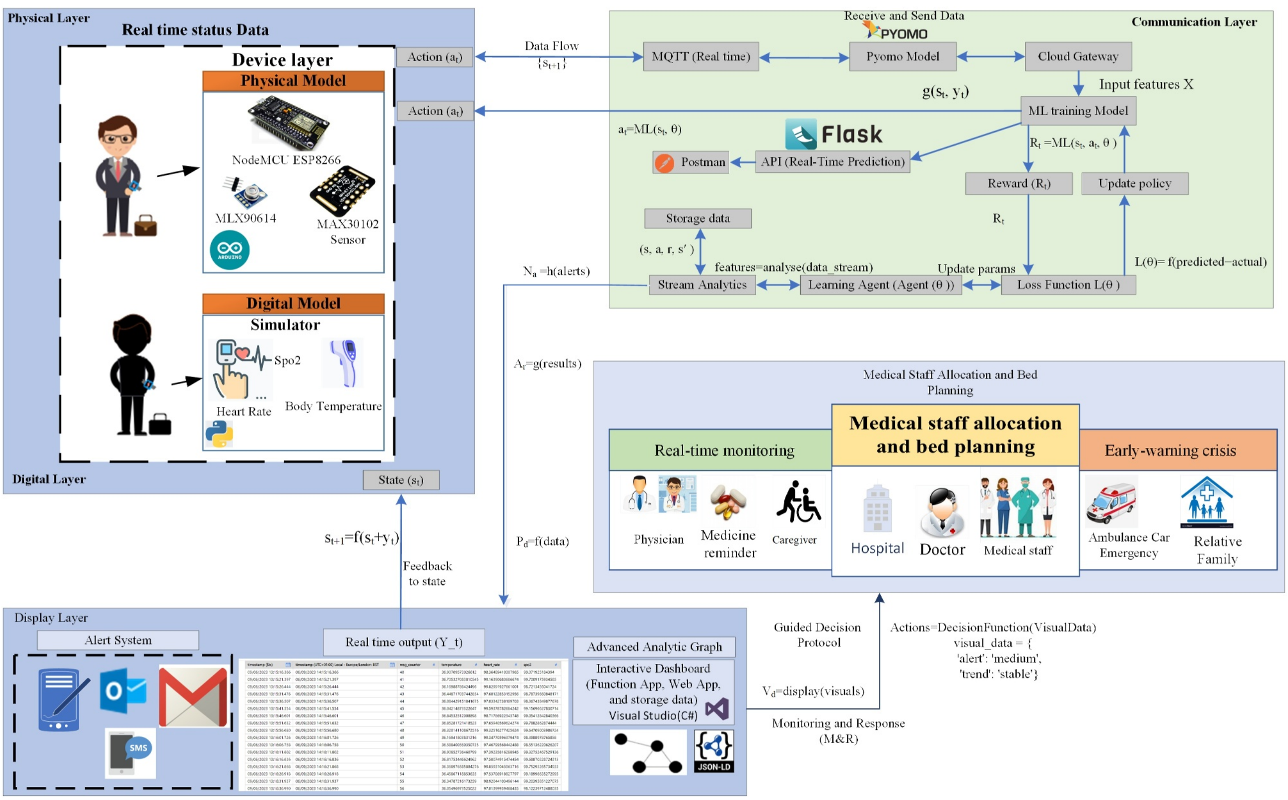


FIGURE 1 Proposed architecture: a digital twin (DT) with the cloud layer, the device layer, the communication layer, and the display layer.

the message queuing telemetry transport (MQTT) protocol, which operates efficiently in low-bandwidth, high-latency IoT environments. Real-time communication is fundamental to the DT (DT) paradigm. Figure 2 illustrates the HR, SpO2 (Blood Oxygen Saturation), and BT readings, which are sampled and sent to the communication layer for further processing. The step-by-step process of sensor integration is outlined as follows:

1. Sensor Selection and Criteria: The sensors used in this framework are the Max30102 for SpO2 and HR measurements and the MLX90614 for BT measurement. The Max30102, chosen for its accuracy in capturing HR and SpO2 data, operates with built-in LEDs and photodetectors. It uses a 1.8 V supply for the sensor and 3.3 V for the LEDs, enabling precise pulse oximetry by detecting changes in light absorption between oxygenated and deoxygenated blood [66, 67]. The MLX90614 sensor was chosen for non-contact temperature measurements, offering accuracy and reliability, particularly in clinical environments where hygiene and patient comfort are priorities. Both sensors were selected due to their compatibility with low-power IoT systems, their precision, and their ability to operate in healthcare monitoring scenarios.
2. Integration of Sensors into the Framework: The Max30102 and MLX90614 sensors are integrated into the framework via the NodeMCU ESP8266, using the I2C communication protocol. The NodeMCU collects data from the sensors at regular intervals and transmits it to the IoT Hub using the MQTT protocol, which ensures reliable data delivery, even under constrained network conditions. The integration process, as shown in Algorithm 1, describes the initialisation of the system, sensor data collection, and transmission of the gathered physiological parameters to the IoT Hub.
3. Data Flow and Real-Time Monitoring: The NodeMCU ESP8266 collects multiple sensor readings (SpO2, HR, and BT), and these are transmitted to the IoT Hub (gateway) for further processing. The IoT Hub handles device-to-cloud (D2C) communication, ensuring system resilience and validation through cloud integration. The NodeMCU indicates successful communication with the cloud via the serial monitor, showing 'OK' upon establishing a connection.

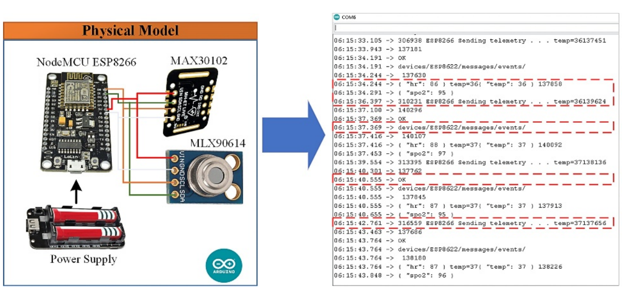


FIGURE 2 Sensor output in the IoT device network.

$$Spo_2 = \frac{\mathcal{A}_{C_{red}} \div \mathcal{D}_{C_{red}}}{\mathcal{A}_{C_{IR}} \div \mathcal{D}_{C_{IR}}}. \quad (1)$$

4. Algorithm and Implementation: The data flow for monitoring patient vitals is implemented in Algorithm 1, where sensor readings are processed and transmitted in real-time to the IoT Hub. The algorithm, implemented using the Arduino integrated development environment (IDE), ensures that the system continuously monitors patients, updating the DT model with real-time data and supporting predictive analytics for patient surveillance.

Algorithm 1. Intelligent Patient Surveillance and Predictive Analytics with DT (Arduino IDE)

Procedure Main:
  Initialisation: Set devices and IoT Hub; set time;
Input: $\mathfrak{D}$(hr, BT, Spo2), $\mathcal{H}_D$(Historical data)
Output:
- $\mathcal{S}_D$ = status of the patient
- DT = Digital Twin model
- $\mathcal{F}_{DT}$ = Digital Twin with Machine Learning

  $\Delta\mathfrak{D}_{DT} = \mathfrak{D}(t_0) - \mathfrak{D}(t_0 - 1)$ # Change in data at time t_0
  $\mathfrak{B} = \mathfrak{B} + 1$ # Calculate number of data points
  if $\mathfrak{B} = 20$ **then**
    $\mathfrak{D} = \max(\mathfrak{D}(n)) - \min(\mathfrak{D}(n))$ # Data difference
    $\mathcal{D}_r = \max(\mathfrak{D}) - \min(\mathfrak{D})$ # Range of data values
    if $\Delta\mathfrak{D} < 0$ **then**
      $D_\xi = \frac{\Delta\mathfrak{D}}{\Delta\mathfrak{D}+\mathfrak{D}} \times \mathcal{D}_r$ # Supplementary data flow rate
    else
      $D_\xi = 0$
      $\mathfrak{D} = 0$
    end if
    if $n \neq 1$ and $\Delta\mathfrak{D} = 0$ **then**
      $\mathcal{M}(c_0) = \mathcal{M}(c_0 - 1) + 1$ # Monitor the patient
    end if
    if $\mathcal{M}(c_0) > 1$ **then**
      $\mathfrak{I}_{df} = \mathcal{D}_r \times \frac{(\mathcal{M}(c_0)-1)}{\mathfrak{I}_s}$ # Data flow rate
    end if
  else
    if $n = 1$ or $\Delta\mathfrak{D} \neq 0$ **then**
      $\mathcal{M}(c_0) = \mathcal{M}(c_0 - 1)$
    end if
  end if
  $\mathcal{M}(c_0)$ continues for the next cycle of monitoring
End of Procedure

4.3 | Digital model layer

The virtual object model aims to provide users with medicine reminders and emergency alerts while monitoring a patient's physiological status in using wearable devices. It involves constructing a participant's history, sensor model, and behavioural model to characterise activities, for example, medication and crisis behaviour. These models can predict future actions and enable evaluation, reasoning, and prediction using rules of associations, restrictions, and deductions, as shown in Figure 1. The DTH model is expressed as follows:

$$DTH = (\mathcal{P}_i,\ \mathcal{S}_m,\ \mathcal{B}_m\ ,\ \mathcal{F}_m,\ \mathcal{N}_c), \tag{2}$$

$\mathcal{P}_i$ includes personal information, for example, name, gender, age and historical data. This information is utilised as a foundation for managing personal health $\mathcal{S}_m$ mostly consists of medical sensor data, for example, HR, SpO2 and BT. This model must design a behavioural model $\mathcal{B}_m$ to characterise the state of a person, a sick person or an elderly person to monitor his status, including the quantity of medicine taken or crisis behaviour, for example, cardiac arrest or respiratory arrest. $\mathcal{N}_c$ describes the Internet connection between the cloud environment and devices. Finally, model evolution is conducted in parallel, and models are calibrated to run synchronously with physical objects. The advanced models provide more precise estimation, optimisation and forecasting of operational process model ($\mathcal{F}_m$).

4.4 | Communication layer

Real-time data exchange between the device layer, the edge layer and the cloud layer is made possible by the communication layer. The protocols that have been implemented in the communication layer are HTTPS and MQTT, as shown in Figure 2. The communication layer ($\mathcal{C}_\mathcal{L}$) includes three parts expressed as follows:

$$\mathcal{C}_\mathcal{L} = (\mathcal{C}_{\mathcal{L}-PD},\ \mathcal{C}_{\mathcal{L}-VD},\ \mathcal{C}_{\mathcal{L}-PV}), \tag{3}$$

where $\mathcal{C}_{\mathcal{L}-PD}$, $\mathcal{C}_{\mathcal{L}-VD}$, and $\mathcal{C}_{\mathcal{L}-PV}$ denote the communication between physical objects and DT data ($\mathfrak{D}_{DT}$), virtual objects and $\mathfrak{D}_{DT}$, and physical and virtual objects, respectively. The $\mathfrak{D}_{DT}$ model includes data from physical and virtual objects. The data source, value, unit, and sample size all play a role in the data that are sent and received throughout each communication.

The model and the interaction between the real-time data inputs and our learning system through a sequence of transformations and updates aimed at optimising action decisions based on rewards. Initially, the state transitions are dictated by

$$s_{t+1} = f(s_t, a_t, e_t), \tag{4}$$

where s_{t+1} is the next state, s_t is the current state, a_t is the current action taken based on the policy, and e_t is an external factor in the environment. The system evolves over time influenced by actions and external factors. Actions are selected using a ML model

$$a_t = ML(s_t, \theta), \tag{5}$$

designed to maximise the perceived rewards. Where a_t is the action decided by the ML model based on the current state s_t and model parameters θ. These rewards are calculated as follows:

$$R_t = ML(s_t, a_t, \theta). \tag{6}$$

This represents the reward calculated by the ML model based on the current state s_t, action a_t, and parameters. Rewards are used to continually refine the model's policies via the update equation

$$\theta_{\text{new}} = \theta_{\text{old}} + \alpha \nabla_\theta J(\theta), \tag{7}$$

ensuring the system learns to respond more effectively to dynamic conditions. Where θ_{new} and θ_{old} are the updated and previous parameters of the model, respectively, α is the learning rate, and $J(\theta)$ is the objective function, typically the expected reward. Additionally, the system's performance is evaluated using a loss function

$$L(\theta) = \left(R_t - \hat{R}_t(\theta)\right)^2, \tag{8}$$

where $L(\theta)$ is the loss function, R_t is the actual reward received, and $\hat{R}_t(\theta)$ is the predicted reward, given the model parameters θ. In the learning agent, the stream analytics (feature extraction)

$$X_t = \text{analyse}(s_t), \tag{9}$$

where X_t represents the features extracted from the state s_t by the stream analytics process. Also, guiding further refinements to model parameters through the learning process

$$\theta_{\text{new}} = \theta_{\text{old}} - \eta \frac{\partial L}{\partial \theta}, \tag{10}$$

the η is the learning rate, and the partial derivative $\frac{\partial L}{\partial \theta}$ represents the gradient of the loss function with respect to the model parameters, thereby achieving a balance between exploration and exploitation in decision-making.

4.5 | Display layer

As shown in Figures 1 and 2, the significance of the Display Layer inside the DT system, which is mostly based on a cloud platform. The fundamental aspect of the layer is the integration of TSI for the purpose of data analytics, as well as storage for data preservation. On the other hand, the Azure Digital Twins

(ADT) explorer specifically caters to the visual exploration and administration of DT. Function apps have become an essential intermediary, facilitating the connection between TSI and the twin graph. This connection allows for efficient data flow and processing using the capabilities of visual studio.

At the core of the architectural framework lies the DT Definition Language (DTDL), which serves to establish the models of the DT, while the twin graph visually represents the interconnectedness between these models. The layer reaches its peak with a wide range of output channels, for example, JSON-LD for the exchange of data, short message service (SMS) for immediate warnings, and a variety of notifications sent via mobile, online, and email interfaces. Also, real-time data from a diverse array of inputs, including HR, SpO2, and BT monitors, are aggregated at a physical device and DT layers. This data is encapsulated within a state s_t, and transitions based on both internal dynamics and external feedback

$$s_{t+1} = f\left(s_t, y_t\right), \tag{11}$$

reflecting the system's adaptive capabilities. Data is processed

$$P_d = \text{process}(\text{data}), \tag{12}$$

analysis is

$$A_r = \text{analyse}(\text{data}), \tag{13}$$

and monitored for critical thresholds to generate alerts

$$N_a = \text{alert}(\text{data}), \tag{14}$$

all of which are displayed in a comprehensive alert system designed to support medical staff in real-time decision-making

$$V_d = \text{display}(\text{visuals}). \tag{15}$$

This integration not only facilitates immediate responses but also enables a continuous feedback loop, enhancing the system's accuracy and responsiveness.

In addition, DT data healthcare services include the physical and digital states of items, as well as information on services and how those two states are fused, as shown below.

$$\mathfrak{D}_{DT} = (\mathcal{D}_{PA}, \mathcal{D}_{DO}, \mathcal{D}_{HS}, \mathcal{D}_F), \tag{16}$$

where $\mathfrak{D}_{DT}$ is DT data from physical and virtual objects, $\mathcal{D}_{PA}$is the physical asset data, $\mathcal{D}_{DO}$ is the digital object data, $\mathcal{D}_{HS}$ is the data from historical data (record in hospital), and $\mathcal{D}_F$ is the data fusion. The equivalent value at the present time, denoted by t_0, can be expressed as follows:

$$\Delta\mathfrak{D}_{DT} += \mathfrak{D}(t_0) - \mathfrak{D}(t_0 - 1). \tag{17}$$

The data processing responsibilities, illustrating data transfer and processing in the proposed architecture, which are explained in Sections 4.2 and 4.3.

5 | FRAMEWORK DEVELOPED ON THE BASIS OF TWIN GRAPH

Digital Twin Definition Language is an open modelling language in ADT that promotes transparency and interoperability. It allows developers to define twins based on telemetry, attributes, and instructions. Digital Twin Definition Language also allows for characterising connections between virtual and physical objects. It requires a common representation of locations, infrastructure, and assets for interoperability and data exchange across domains, as shown in Figure 1.

In the present work, DTDL uses JSON-LD, an open language akin to JSON. The DTH platform allows remote monitoring and health tracking. It is scalable for smart devices and patients. An IoT hub sends patient monitoring data to the cloud through a twin-graph platform. Serverless Functions apps reduce code, infrastructure administration, and expenses. Within the broader context of the research, Algorithm 2 was implemented to synchronise data between IoT devices and the DTs environment. The following key steps were involved.

1. At the outset of the algorithm, necessary libraries are imported, including the DTs software development kit for NET, cloud Identity for authentication, as well as additional utilities for parsing and managing IoT device data. These libraries are crucial for establishing a secure connection to DTs, facilitating authentication and enabling the manipulation of DT data.
2. The environment variables are mainly ADT_URL, which stands in place for the DTs instance endpoint uniform resource locator (URL). The URL is important because it directs the algorithm to the right DT service, hence applying updates to the right environment with no confusion.
3. This shields the algorithm from being initiated in the environment variable case where ADT_URL is not configured; all such cases are logged. If ADT_URL is undefined, then all parameters should be set up; otherwise, an error log is generated.
4. Credential objects are crafted through Default Cloud Credential method for smooth authentication in the suite of cloud services. After that, the connection is built with DTs through the use of these newly crafted credentials to establish a secure and strong link with the DT client object.
5. Next, the UpdateDigitalTwin function is invoked using the device's unique identifier and a dictionary of updated data, which accurately reflects the latest sensor information on the DT. This ensures that the DT properly holds the latest status of the physical device.

Algorithm 2. Digital Twin Data Update

```
Start
Import necessary libraries
Input: ℰ (EventGridEvent), ℒ (log)
```

```
Output: Status of the update
A_U ← getenv('ADT_URL')
if httpClient is true then
     curl_easy_setopt(httpClient)
end if
if A_U = null then
   L. LogError("Error: 'ADT_URL' is not set")
 else
   C← Credentials() % Default cloud credential
   DTC← (A_U, C) % Digital Twin Client
   L. LogInformation("Connected to ADT") % Log the connection status
   if E ≠ null and E. Data ≠null then
   DM ← JObject.Parse (E. Data) % Device Message
     DI ← DM. GetValue('IoTHub Id').ToString() % Device ID
     DS ← DM. GetValue('DeviceData').ToString() % Device Data
     L. LogInformation("Device ID: " + DI ", DeviceData: " + DS) % Log device information
     UD ← new Dictionary<string, object>() % Update Twin Data
     UD. Add("DeviceData", DS) % Add device data to dictionary
     DTC.UpdateDigitalTwin(DI, UD) % Send update to Digital Twin
   end if
 end if
 End of function
```

Digital Twin Definition Language was used not only to define telemetry data streams but also to describe the relationships between IoT devices, sensors, and virtual models. These definitions were fed into the twin graph to model real-time interactions between physical and virtual entities. The twin graph provided a structured view of how physical assets (e.g. sensors) communicated with the DT system, ensuring scalability and the easy integration of new devices.

Additionally, the algorithm is designed to efficiently manage a high volume of events, reducing latency and guaranteeing timely updates to DTs, thus addressing performance and scalability concerns. The scalability of our system, as supported by the communication protocols listed in Table 2, demonstrates the capability of our architecture to handle increased traffic and data load efficiently. Protection for sensitive data during transmission and access is ensured through encrypted communications and secure authentication methods, upholding compliance with privacy standards and regulations.

Moreover, seamless integration with the system's architecture is achieved by Algorithm 2, enabling dynamic updates to DTs and mirroring changes in the physical world. Crucially, this algorithm works in conjunction with IoT Hub for device management and DTs for modelling and simulating real-world environments, becoming indispensable for the system's monitoring, analysis, and response capabilities as illustrated in the PoC, referenced in Subsection 7.2.

In Figure 3, a patient is connected to two sensors using a knowledge graph, allowing for multiple sensors and monitoring. An integrated environment is created, monitoring SpO2, HR, and BT in the cloud. Machine learning is used to predict patients' future states, demonstrating the development of a novel architecture with multiple applications. In Figure 4, the expanded framework is illustrated in ADT explorer,

TABLE 2 System parameters.

Parameter	Value
Scenario	Indoor/outdoor
Channel band/Bandwidth	2.4 GHz/20 MHz
BS/UE Tx power	20/15 dBm
Traffic	MQTT, sensor data
Packet size	1024 bytes
Data transmission frequency	2–8 Hz
Communication protocols	MQTT over SSL/TLS, I2C
Security	TLS with X.509 certificates, hybrid encryption
Solver	GLPK (linear programming solver)
Integrity verification	Merkle tree validation
Authentication	Certificate-based authentication

Abbreviations: BS, base station; GLPK, GNU linear programming kit; SSL, secure sockets layer; TLS, transport layer security; UE, user equipment.

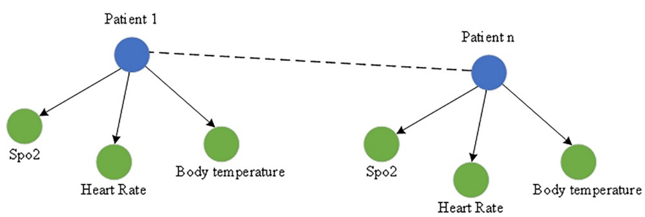


FIGURE 3 Structure of digital twin (DT) using the twin graph.

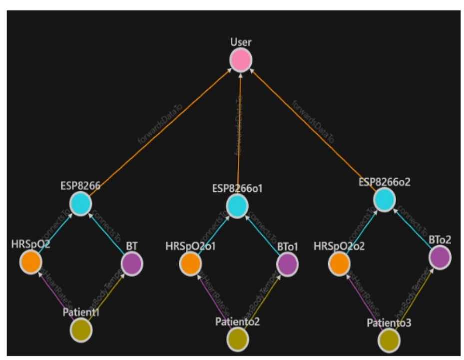


FIGURE 4 Implementation of digital twin (DT) in Azure Digital Twins (ADT) explorer.

showing the monitoring capabilities for BT, SpO2, and HR across multiple patients. Connections are established between each patient and their respective sensors via an ESP8266 controller, allowing data transmission to a user monitoring entity. The design is scalable to accommodate various patient demographics, capturing attributes.

In Section 6, ML is used to develop a novel architecture with multiple applications. A DT instance is created, authenticated via roles, and fed data from IoT edge devices through representational state transfer (REST) application programming interface (APIs). The operator assigns roles to members and doctors, while ADT enables querying, model updates, and data visualisation through the explorer. Models are authored in DTDL and stored as JSON-LD files.

6 | IMPLEMENTATION SETUP

Digital twin platform for intelligent healthcare systems was developed, utilising cloud computing, wearable devices, data analytics, and ML. This platform enabled remote patient monitoring by creating virtual patient replicas. The physical and virtual elements of the system were implemented, with the physical element's functionality outlined in Algorithm 1 and the virtual element's data transfer, processing, and decision-making described in Algorithm 2. Machine learning techniques were applied, and seven different algorithms were evaluated to select the optimal predictive model.

Figure 5 presents the sequence diagram, depicting the interactions between system components and the data flow during telemetry processing. The diagram illustrates the generation, validation, and handling of telemetry data, supporting bidirectional communication for healthcare monitoring. Telemetry was received, validated, and processed, followed by real-time visualisation and data archiving. The ML model deployed in the system enhanced decision-making, while automated alerts were triggered based on the telemetry data.

Table 2 outlines the system parameters, highlighting security protocols (e.g. transport layer security [TLS] with X.509 certificates and hybrid encryption) and data transmission methods such as MQTT over secure sockets layer/TLS. Adjustable data transmission frequencies (2–8 Hz) allowed for flexibility in patient monitoring. The GLPK solver optimised telemetry frequency, improving system responsiveness, while Merkle Tree validation ensured data integrity.

Algorithm 3. Telemetry Optimisation and Resource Monitoring with Pyomo

```
Procedure: Telemetry Transmission and Optimisation
Initialisation:
    Connect devices to IoT Hub
    Initialise Pyomo model for optimisation (set frequency, priority)
    Measure initial network and CPU usage
Step 1: Data Collection and Telemetry Generation
    Generate telemetry data (normal and abnormal) for heart rate, SpO2, and temperature
    Log telemetry data and send to cloud
Step 2: Pyomo Model Optimisation
    Define variables for telemetry frequency and priority in the Pyomo model
    Set constraints on frequency and priority ranges
    Solve the Pyomo model to minimise latency L
Step 3: Transmit Telemetry and Measure Resource Usage
    Send telemetry data to the cloud
    Measure CPU usage and network usage before and after telemetry transmission
    Calculate data sent during transmission
Step 4: Log Resource Efficiency
    Log CPU usage, network usage, and telemetry data in the resource efficiency log
Step 5: Update Telemetry Transmission Frequency
    Adjust telemetry transmission frequency based on Pyomo model results
    Update frequency for the next telemetry transmission cycle
Repeat the Process:
    Continue generating telemetry, optimising transmission, and monitoring resources in the next cycle
End of Procedure
```

6.1 | Pyomo model with Digital Twin

The Pyomo model was integrated into the DT framework to optimise telemetry transmission and manage resources. As illustrated in Figure 1, Pyomo's linear programming functionality dynamically adjusted telemetry frequency and prioritisation in response to network conditions, reducing latency and maintaining efficient CPU and network performance. Algorithm 3 describes the process of telemetry optimisation and resource monitoring.

6.2 | Data acquisition

The ML model was trained to predict cardiovascular disease, hyperthermia, and abnormal SpO2 using data from the MIMIC-III clinical database [68]. Volunteers' data were collected to validate the model, with vital metrics labelled according to medically accepted ranges. Abnormal readings were classified if they exceeded these thresholds. Body temperature ranged from 36.5 to 37.3°C; heartbeats per minute ranged from 60 to 100; SpO2 should be at least 95% [69, 70].

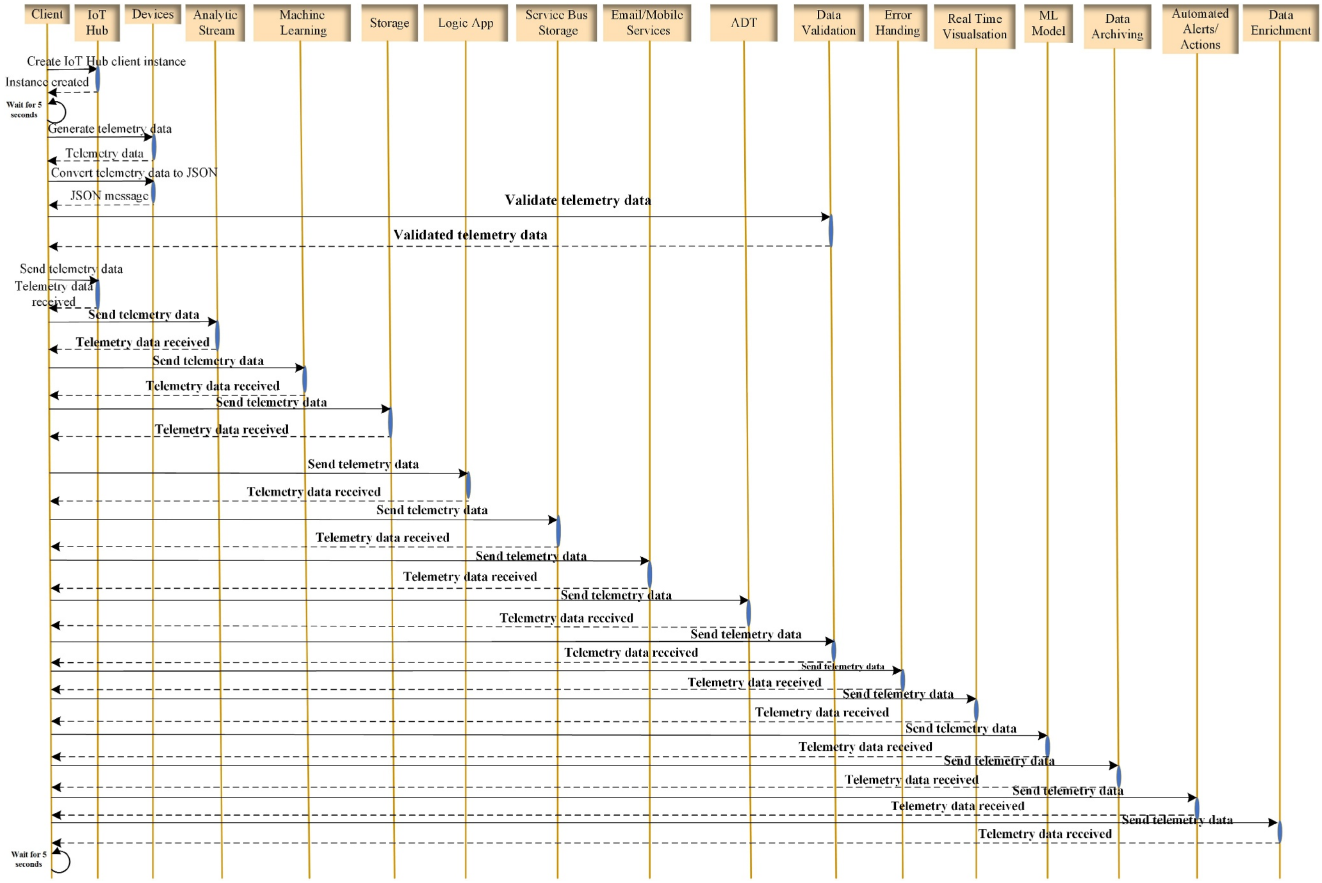


FIGURE 5 Model of the digital twin (DT) sequence diagram.

6.3 | Data preparation

Data were analysed using statistical methods, histograms, and outlier detection. Missing data were handled using imputation techniques, and feature selection reduced the input size. The Pearson correlation coefficient was used to examine variable dependence, as described in Algorithm 4. Outliers were removed, and the dataset was normalised to ensure accuracy and consistency.

Algorithm 4. Data preprocessing functional element

```
Input: BT, HR, Spo2
Output: Results
  Results ← list()
for feature ∈ [BT, HR, Spo2] do
  if isNormalised(feature) is True then
    removeOutliers(feature)
    resize(feature)
    format(feature)
    Results.append(feature)
  else
    continue
  end if
end for
return Results
```

6.4 | Data standardisation and model evaluation

The StandardScaler technology was used to standardise the data. In addition, 75% of the data was utilised for training purposes, whereas only 25% was used for actual testing. In this modelling process, we evaluate seven widely used ML classification algorithms and compare their performance [71, 72]: Random forest, Gaussian Naïve Bayes, logistic regression (LR), K-nearest neighbours, decision tree (D-Tree), XGBoost and support vector machine.

In Figure 6, the structure of the XGBoost-based model design is presented. The model was optimised using the following hyperparameters: a learning rate of 0.01, a maximum depth of 3, and 100 estimators. These values were selected after performing grid search to tune the hyperparameters, ensuring that the model achieved the best possible predictive performance. As a result, the XGBoost model delivered high accuracy and efficient real-time predictions of health outcomes.

7 | PROOF OF CONCEPT

7.1 | Integration and communication of IoT health devices

Figure 2 presented the integration of IoT devices within a physical model, demonstrating health data transmission. The serial monitor captured the output from an ESP8266 module, including telemetry such as HR and BT. These data points are timestamped and transmitted to an IoT node, as indicated by the dashed red lines. The physical model depicted the configuration of connected devices: the NodeMCU ESP8266 microcontroller, MAX30102 pulse oximeter, and MLX90614 infrared thermometer, representing a patient. This setup demonstrated the practical application of IoT in health monitoring, aligning with earlier discussions in Subsection 4.5.

In addition to SpO2, HR, and BT, the framework is designed to support additional sensors for monitoring blood pressure, motion, and glucose levels. The adaptable architecture allowed for the integration of these sensors, ensuring real-time transmission and analysis through cloud-based systems, providing comprehensive and personalised healthcare monitoring in both clinical and home settings.

Real-time monitoring has been designed to be highly effective in clinical and home environments. The system, utilising IoT devices like ESP8266 microcontrollers, transmitted health data to the cloud in real time. This allowed physicians, caregivers, and family members to receive alerts and insights promptly. Predictive analytics powered by ML (e.g., XGBoost) provided forecasts with 98% accuracy, supporting decision-making for healthcare professionals.

In home settings, the portable IoT devices and lightweight sensors ensured patients were monitored remotely without disrupting daily activities. The cloud-based system provided real-time health updates and alerts, enabling timely care even in remote areas.

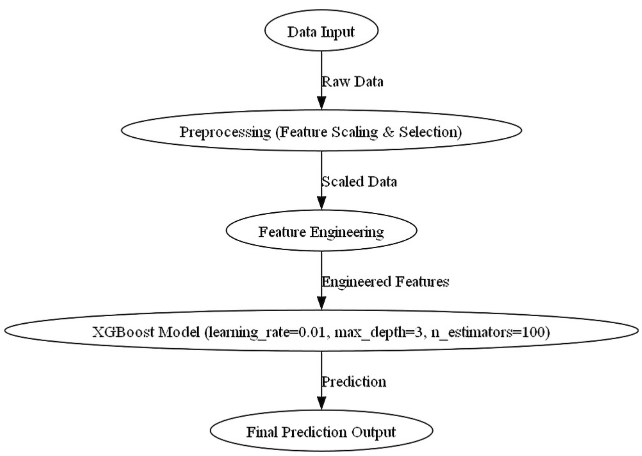


FIGURE 6 Structure of the XGBoost-based model design. The flowchart illustrates the stages from data input to final prediction, detailing the feature engineering and machine learning (ML) process.

7.2 | Schematic and data flow analysis in Digital Twin systems

Figure 4 displayed the configuration of the DT network within the DTs explorer, illustrating the flow of patient data through connected sensors. Data from BT and HR sensors converged at the ESP8266 microcontroller, demonstrating its role in biometric data aggregation and transmission. Telemetry and attributes from IoT devices were defined by the DTDL schema and streamed in real time for analysis using the twin graph platform.

The system leveraged Time Series Insights (TSI) for processing time-series data and detecting anomalies. The flow of patient data was visualised and traced through the twin graph, ensuring accuracy in both virtual and physical states, with trend analysis provided by TSI. Figure 7 depicted the schematic design of the health monitoring system, where patient-specific data and sensor interfaces are integrated into a modular DT environment for comprehensive health management. Figure 8 presented live data from health-monitoring sensors (MAX30102 for SpO2 and HR and MLX90614 for BT). The metrics are updated dynamically, validating the system's capacity for continuous patient monitoring.

7.3 | Function execution for IoT Hub and digital twins integration

Figure 9 illustrated the invocation logs of a function app, acting as an intermediary between the IoT Hub and DTs. The logs confirmed that telemetry data was processed upon receiving an event trigger, with details of device IDs, authentication, and timestamps. The telemetry data are decoded and used to update the DT's state.

7.4 | Temporal analysis and validation of biometric data streams

Figure 10 showed a tabulated excerpt of biometric data from TSI, representing timestamped readings of BT, HR, and SpO2.

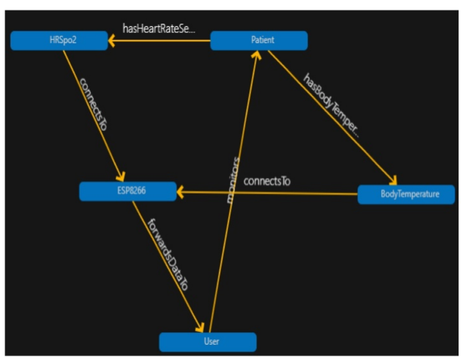


FIGURE 7 Structure of the digital twins (DTs) model for health monitoring.

This tabulation validated the structure of data used for temporal analysis, highlighting continuous telemetry captured from patient monitoring sensors. Figure 18 presented a graphical representation of biometric readings over time, visualising trends for HR, SpO2, and BT. The fluctuations in the data streams were plotted, providing insights into the patient's physiological status and validating the system's monitoring capabilities.

7.5 | Privacy and ethical considerations in Digital Twin healthcare systems

The integration of DTs and cloud computing for healthcare monitoring raised privacy and ethical concerns. Strong encryption is used to protect patient data during transmission between IoT devices, DTs, and the cloud. End-to-end encryption secured the data both in transit and at rest, while secure authentication protocols ensured that only authorised healthcare providers can access the information.

Patient consent is prioritised through opt-in digital agreements, ensuring that patients retained control over their data. Role-based access control is implemented to enforce fine-grained permissions. Our framework adhered to privacy-by-design principles to minimise data exposure and ensure transparency in data processing.

Building on previous research focused on DT security and secure task offloading mechanisms in edge computing [73], this framework further enhanced cybersecurity for DTs in healthcare systems, addressing specific challenges related to data privacy and security.

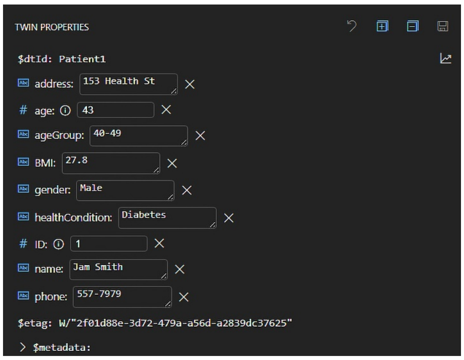
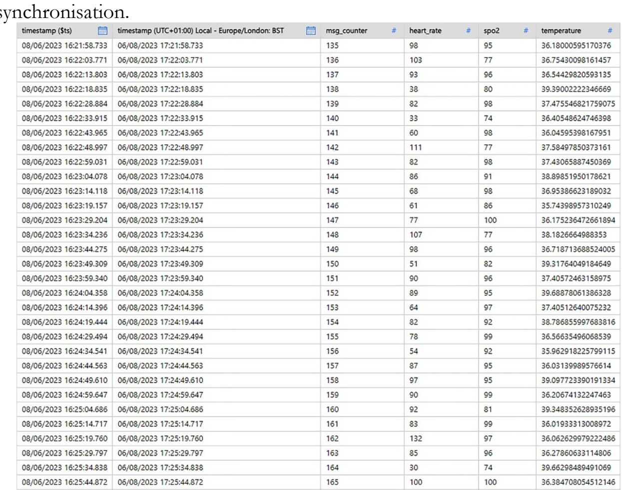


FIGURE 8 Live data feed from health monitoring sensors.

FIGURE 9 Function invocation logs for IoT and digital twins (DTs) synchronisation.



timestamp ($ts)	timestamp (UTC+01:00) Local - Europe/London: BST	msg_counter	heart_rate	spo2	temperature
08/06/2023 16:21:58.733	06/08/2023 17:21:58.733	135	98	95	36.18000595170376
08/06/2023 16:22:03.771	06/08/2023 17:22:03.771	136	103	77	36.75430098161457
08/06/2023 16:22:13.803	06/08/2023 17:22:13.803	137	93	96	36.54429820593135
08/06/2023 16:22:18.835	06/08/2023 17:22:18.835	138	38	80	39.39002222346669
08/06/2023 16:22:28.884	06/08/2023 17:22:28.884	139	82	98	37.475546821759075
08/06/2023 16:22:33.915	06/08/2023 17:22:33.915	140	33	74	36.40548624746398
08/06/2023 16:22:43.965	06/08/2023 17:22:43.965	141	60	98	36.04595398167951
08/06/2023 16:22:48.997	06/08/2023 17:22:48.997	142	111	77	37.58497850373161
08/06/2023 16:22:59.031	06/08/2023 17:22:59.031	143	82	98	37.43065887450369
08/06/2023 16:23:04.078	06/08/2023 17:23:04.078	144	86	91	38.89851950178621
08/06/2023 16:23:14.118	06/08/2023 17:23:14.118	145	68	98	36.95386623189032
08/06/2023 16:23:19.157	06/08/2023 17:23:19.157	146	61	86	35.74398957310249
08/06/2023 16:23:29.204	06/08/2023 17:23:29.204	147	77	100	36.175236472661894
08/06/2023 16:23:34.236	06/08/2023 17:23:34.236	148	107	77	38.1826664988353
08/06/2023 16:23:44.275	06/08/2023 17:23:44.275	149	98	96	36.718713688524005
08/06/2023 16:23:49.309	06/08/2023 17:23:49.309	150	51	82	39.31764049184649
08/06/2023 16:23:59.340	06/08/2023 17:23:59.340	151	90	96	37.40572463158975
08/06/2023 16:24:04.358	06/08/2023 17:24:04.358	152	89	95	39.68878061386328
08/06/2023 16:24:14.396	06/08/2023 17:24:14.396	153	64	97	37.40512640075232
08/06/2023 16:24:19.444	06/08/2023 17:24:19.444	154	82	92	38.786855997683816
08/06/2023 16:24:29.494	06/08/2023 17:24:29.494	155	78	99	36.56635496068539
08/06/2023 16:24:34.541	06/08/2023 17:24:34.541	156	54	92	35.962918225799115
08/06/2023 16:24:44.563	06/08/2023 17:24:44.563	157	87	95	36.03139989576614
08/06/2023 16:24:49.610	06/08/2023 17:24:49.610	158	97	95	39.097723390191334
08/06/2023 16:24:59.647	06/08/2023 17:24:59.647	159	90	99	36.20674132247463
08/06/2023 16:25:04.686	06/08/2023 17:25:04.686	160	92	81	39.348352628935196
08/06/2023 16:25:14.717	06/08/2023 17:25:14.717	161	83	99	36.01933313008972
08/06/2023 16:25:19.760	06/08/2023 17:25:19.760	162	132	97	36.062629979222486
08/06/2023 16:25:29.797	06/08/2023 17:25:29.797	163	85	96	36.27860633114806
08/06/2023 16:25:34.838	06/08/2023 17:25:34.838	164	30	74	39.66298489491069
08/06/2023 16:25:44.872	06/08/2023 17:25:44.872	165	100	100	36.384708054512146

FIGURE 10 Biometric data table from time series insights.

8 | RESULTS AND DISCUSSIONS

8.1 | Model comparative analysis

The final ML model was selected after a comprehensive comparison of multiple models. Key metrics, including accuracy ($\mathcal{A}$), were assessed alongside computational time. Accuracy, calculated using true positive ($\mathcal{TP}$), true negative ($\mathcal{TN}$), false positive ($\mathcal{FP}$), and false negative ($\mathcal{FN}$), is shown below:

$$\mathcal{A}(\%) = \frac{\mathcal{T}_{\mathcal{P}} + \mathcal{T}_{\mathcal{N}}}{\mathcal{T}_{\mathcal{P}} + \mathcal{T}_{\mathcal{N}}} + \mathcal{F}_{\mathcal{P}} + \mathcal{F}_{\mathcal{N}} \times 100. \tag{18}$$

Figure 11 compares the accuracies of the seven classifiers tested, with XGBoost exhibiting the highest accuracy. All models achieved over 82%, demonstrating the robustness of the methodology. Figure 12 provides confusion matrices for each classifier, illustrating the distribution of correct and incorrect predictions.

8.2 | Analysing machine learning metrics and model selection

Several performance metrics, including Precision ($\mathcal{P}$), Recall ($\mathcal{R}$), and F1 Score ($\mathcal{F}$), were evaluated to assess model efficiency in healthcare applications.

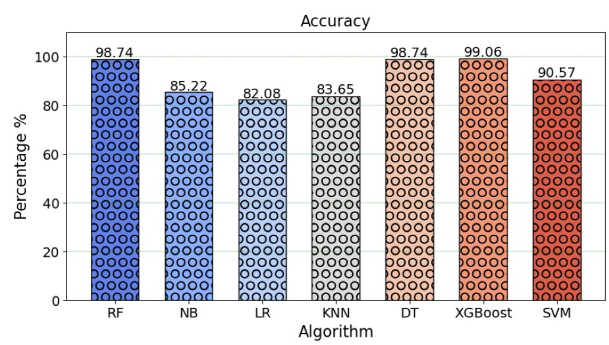


FIGURE 11 Comparison of the accuracy of used models.

Precision ($\mathcal{P}$), defined as the ratio of correctly predicted positive observations to total predicted positive observations, is calculated as follows:

$$\mathcal{P} = \frac{\mathcal{T}_{\mathcal{P}}}{\mathcal{T}_{\mathcal{P}} + \mathcal{F}_{\mathcal{P}}}. \tag{19}$$

Recall ($\mathcal{R}$), measuring correctly predicted positive observations out of all actual positives, is calculated as follows:

$$\mathcal{R} = \frac{\mathcal{T}_{\mathcal{P}}}{\mathcal{T}_{\mathcal{P}} + \mathcal{F}_{\mathcal{N}}}. \tag{20}$$

The F1 Score ($\mathcal{F}$), representing the harmonic mean of Precision and Recall, is given by

$$\mathcal{F} = 2 \times \frac{(\mathcal{R} \times \mathcal{P})}{\mathcal{R} + \mathcal{P}}. \tag{21}$$

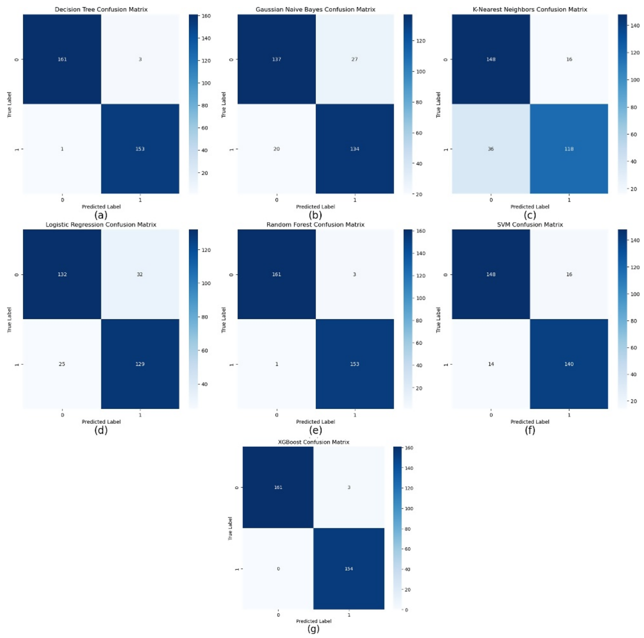


FIGURE 12 Confusion matrices: (a) D-Tree, (b) GNBs, (c) K-nearest neighbours (KNN), (d) logistic regression (LR), (e) Random forest (RF), (f) support vector machine (SVM), (g) XGBoost.

Figure 13 compares Precision, Recall, and F1 Score for all models, with XGBoost achieving the best balance between these metrics.

Figure 14 presented the receiver operating characteristic (ROC) curves of all models, indicating that XGBoost was observed to achieve an area under the curve (AUC) value close to 1, signifying excellent performance in distinguishing between classes. Additionally, Figure 15 compares the computational times of the models, showing that XGBoost offers a reasonable balance between accuracy and efficiency compared to other models.

Table 3 depicted cross-validation (cv = 5) that was conducted to ensure robustness and reliability of the models. The XGBoost model exhibited the highest cross-validation accuracy (CVA) of 99.58%, which confirmed its strong generalisability and its ability to avoid overfitting, in contrast to simpler models such as LR.

Table 4 presented a more rigorous cross-validation (cv = 10) conducted, with the XGBoost model maintaining superior performance, demonstrating stability and consistent accuracy, with a CVA of 99.58%. This emphasises that the XGBoost model is resilient to changes in the training data, making it suitable for deployment in dynamic healthcare environments.

Table 5 illustrated that cross-validation with cv = 20 was conducted for a more exhaustive validation. The XGBoost

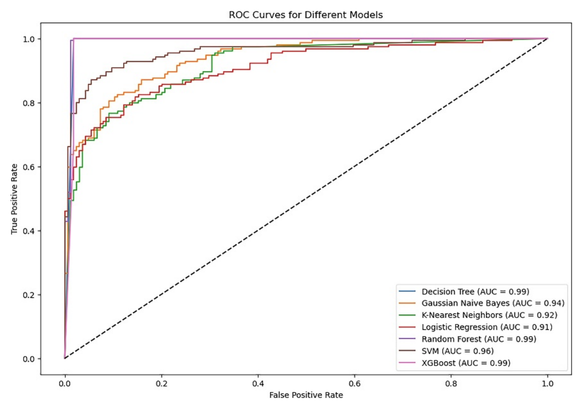


FIGURE 14 ROC curve of seven algorithms. ROC, receiver operating characteristic.

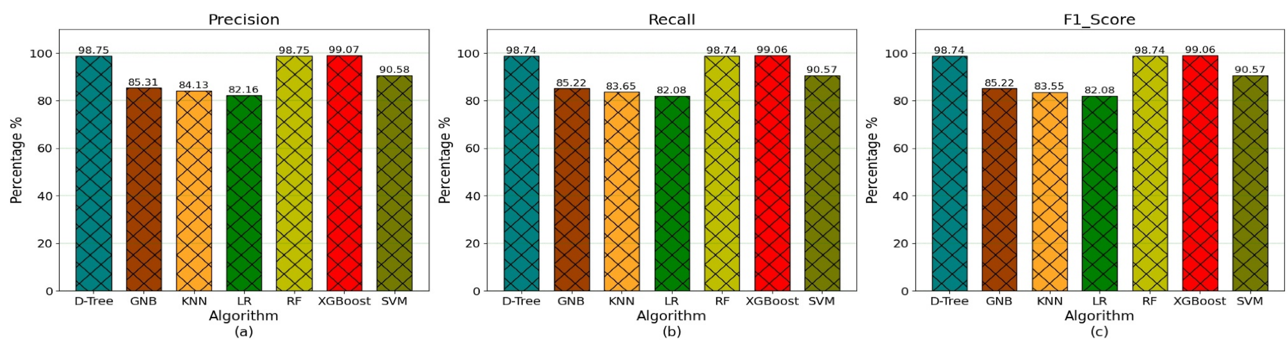


FIGURE 13 Seven algorithms are compared in terms of (a) Precision, (b) Recall and (c) F1 score.

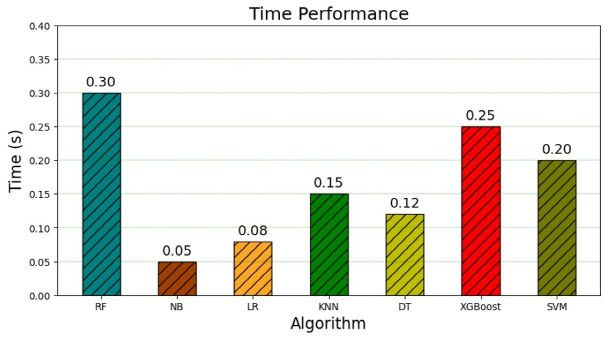


FIGURE 15 Comparison of computational times of seven algorithms.

TABLE 3 Summary of results with cross-validation (cv = 5).

Model	CVA (± std)	Test accuracy
D-tree	0.9916 ± 0.0042	0.9874
GNB	0.8971 ± 0.0143	0.8522
KNN	0.8634 ± 0.0146	0.8365
LR	0.8550 ± 0.0149	0.8208
RF	0.9958 ± 0.0039	0.9874
SVM	0.9149 ± 0.0174	0.9057
XGBoost	0.9958 ± 0.0039	0.9906

TABLE 4 Summary of results with cross-validation (cv = 10).

Model	CVA (± std)	Test accuracy
D-tree	0.9905 ± 0.0099	0.9874
GNB	0.9013 ± 0.0215	0.8522
KNN	0.8760 ± 0.0252	0.8365
LR	0.8487 ± 0.0380	0.8208
RF	0.9947 ± 0.0071	0.9874
SVM	0.9139 ± 0.0203	0.9057
XGBoost	0.9958 ± 0.0070	0.9906

TABLE 5 Summary of results with cross-validation (cv = 20).

Model	CVA (± std)	Test accuracy
D-tree	0.9895 ± 0.0125	0.9874
GNB	0.9013 ± 0.0326	0.8522
KNN	0.8790 ± 0.0402	0.8365
LR	0.8508 ± 0.0510	0.8208
RF	0.9937 ± 0.0096	0.9874
SVM	0.9212 ± 0.0425	0.9057
XGBoost	0.9958 ± 0.0084	0.9906

model again outperformed all other models with a 99.58% accuracy, indicating minimal variability in performance.

The XGBoost model was selected based on several important considerations.

1. Performance Metrics: As shown in Figures 13 and 15, XGBoost consistently outperformed other models across all key metrics (Accuracy, Precision, Recall, F1 Score, and ROC AUC). The model demonstrated particularly high CVA of 99.58% and a near-perfect ROC AUC value, highlighting its effectiveness in distinguishing between positive and negative health states.
2. Speed and Efficiency: XGBoost exhibited reasonable computational time (Figure 15), efficiently handling high-dimensional healthcare data without compromising speed. This balance between performance and computational cost makes it suitable for real-time healthcare monitoring, where timely predictions are essential for patient care.
3. Handling Imbalanced Data: Healthcare datasets often have class imbalances, with significantly fewer positive cases of certain health conditions compared to negative ones. XGBoost's ability to optimise both Recall and Precision enables effective management of imbalanced datasets, minimising both false positives and false negatives, which is critical in healthcare to avoid misdiagnosis and ensure patient safety.

Once selected, the XGBoost model was deployed using cloud services, including logic apps, function apps, and stream analytics, facilitating real-time access for medical professionals, caregivers, and patients alike.

8.3 | Comparative analysis of DTH technologies

Table 6 presents a comparative analysis of various research endeavours or undertakings with respect to their technological frameworks, case study designs, practical implementations, employment of data visualisation and analysis techniques, latency factors, and cost implications. A comparison of the prior literature with our proposed article demonstrates how our research has incorporated and addressed unfinished work from previous studies.

Additionally, our study proposes further work to be undertaken in the future. Our article focused on the expenses associated with various devices used in research, with a particular emphasis on identifying the most cost-effective option. The cloud latency was evaluated, yielding an improved performance of 20–25 milliseconds, demonstrating a significant advancement compared to earlier research. Additionally, the system provided thorough monitoring of outcomes and patient status, facilitated by the integration of TSI and ADT.

Table 7 provides a concise comparison of key performance metrics across different frameworks, including our proposed model, highlighting metrics such as accuracy, response time, telemetry transmission latency, and resource utilisation efficiency. Our proposed framework demonstrates superior performance with a 98% accuracy, a 52% reduction in response time, and a 32% improvement in telemetry latency compared to the other frameworks. Additionally, it optimises CPU usage and provides continuous patient monitoring with high user

TABLE 6 A comparative summary of digital twin (DT) applications in healthcare.

Ref. No.	UT[a]	Case study	Practical work	RTD[b]	DV&A[c]	LAT[d]	Cost
[40](2023)	DT, IoT, DL	Disease detection and smart medical service	Analysis of clinical experimental data	Yes	Yes	–	–
[37](2021)	DT, FEM[f], bidomain model, and 12-lead ECG	Clinical 12-lead ECGs for cardiac electrophysiology DT	MRI, ECG, FEM simulations, parameter adjustment, electrode placement localisation	No	No	–	Yes
[41](2019)	DT, IoT, cloud computing	Elderly healthcare services	Emergency and regular patient simulation with hospital wards	Yes	Yes	–	Yes
[42](2023)	DT, decentralised learning with blockchain.	Industrial ecology learns through data and resource sharing	Experiments and simulations	Yes	Yes	LPT[g]	Yes
[49](2023)	DT, 3-D modelling, DL	Smart clothing system	Hardware–software integration.	Yes	Yes	–	–
[9](2021)	DT, IoT, DL, ML	DT and IoT could revolutionise healthcare	DT-based intelligent context-aware healthcare system proposal and implementation	No	No	–	Yes
[10](2022)	DT, CanTwin	DT technology for workplace virus control by social distance	CanTwin is presented as a practical example of an industry case study	Yes	Yes	4 s	–
[55](2022)	DT, cloud computing	VR cloud framework for interactive DT	Proposing an interactive VR DT cloud infrastructure	–	Yes	Low LAT	–
[56](2023)	DT, IIoT	Precision DT construction.	Concurrent end-to-end synchronisation and multi-attribute data resampling for accurate and efficient DT production.	–	No	–	Yes
[74](2024)	DT, IoT, WBAN	Elderly healthcare monitoring using WBANs	Real-time health monitoring with a focus on reducing packet drop ratios	Yes	Yes	–	M[m]
Proposed work	**DT, IoT, cloud computing, ML, and visualisation**	**Real-time monitoring with integrated virtual and physical data**	**Patient data analysis, condition prediction, alerts, and emergency simulation** via **IoT-cloud framework**	**Yes**	**Yes**	**Low (20–25 ms)**	**Low cost**

Note: The bold values in the "Proposed work" row were used to emphasize the distinctive and critical strengths of the proposed framework compared to others in the table.

Abbreviations: DV, digital visualization; FEM, finite element method; LAT, latency; LPT, low processing time; RTD, real-time data.

[a]Utilised Technology.

[b]Real Time.

[c]Visualisation and Analysis.

[d]Latency.

[f]Finite Element Method.

[g]Less processing Time.

[m]Moderate.

comfort, showcasing its advantage in real-time healthcare applications.

8.4 | Flask deployment for model serving

To facilitate real-time health predictions based on the XGBoost model, we developed a REST API using the Flask framework. The Flask application exposes an endpoint that accepts physiological parameters, performs inference using the trained XGBoost model, and returns the predicted health status along with key performance metrics. The implementation of the Flask code for this model deployment is shown in Figure 16.

Additionally, Postman was used to test the API deployment, as illustrated in Figure 17. A request containing health parameters was sent to the model deployed via Flask, and the real-time prediction metrics, including accuracy, precision, recall, and F1-score, were returned in the response. This demonstrated the successful deployment and real-time prediction capabilities of the system.

8.5 | Web portal

The web application was created to facilitate the effective visualisation of patient data and follow-up by doctors, caregivers, and relatives. The site aims to create DT using the cloud, containing all the features needed to establish a healthcare DT by employing ADT and ML, as well as storing and processing recorded data. This cloud also provides the app logic through which alert messages can be sent via SMS or email to those monitoring the patient. By displaying the patient's physical and physiological data and generating predictions from an ML model, the patient's condition can be diagnosed quickly.

In addition, as shown in Figure 18(a) and on the website, the authors successfully tested the prototype by uploading the detected parameters to the cloud. The digital model was developed based on sensor specifications, recorded data, historical data, and data transmission to the cloud. By utilising Time Series Insights (TSI), a comparison was made between physical and digital objects, revealing a significant degree of resemblance

TABLE 7 Performance comparison of various healthcare monitoring frameworks.

Metric	Proposed framework	[59] 2020	[63] 2023	[75] 2024	[76] 2025	[77] 2024	[78] 2025	[79] 2025	[80] 2025
$\mathcal{A}$	98%	95%	92%	–	95%	–	–	–	–
M1	↓ 52%	50 ms	65 ms	↓ 15%–20%	↓ 20%	↓ 15%–20%	↓ 18%	↓ 10%	↓
M2	↓ 32% (20–25 ms)	20 ms	25 ms	↓ 10%–15%	↓ 15%–25%	↓ 10%–15%	↓ 18%	–	↓
M3	↑ 18%	15%	10%	↑	↑ 20.97%	↑ Throughput	↑ 18%	↓ 5.35%[4]	↑
M4	Adjustable (2–8 Hz)	4–10 Hz	3–7 Hz	Adjustable	Adjustable	Adjustable	Adjustable	–	–
M5	Stable, optimised	Optimised	High[1]	Optimised	Optimised	Optimised (with DRL)	Optimised (A3C)	Optimised	Optimised
M6	98%	96%	90%	97%	–	–	–	↓ 15.77%[2]	–
M7	Continuous	Continuous	Continuous	–	Distributed	–	Distributed	–	–
M8	High	Moderate	High[3]	High	High	High	High	High	High

Note: Metric: M; M1: Response Time; M2: Telemetry Transmission Latency; M3: Resource Utilisation Efficiency; M4: Data Transmission Frequency; M5: CPU Usage; M6: Real-Time Accuracy; M7: Patient Monitoring (SpO2, HR, BT); M8: User Comfort; Requires optimisation[1]; Improved with reduced SLA violation rate[2]; Non-invasive sensors[3]; Improved due to reduced energy consumption by 5.35%[4]; ↓: To indicate reduction; ↑: For improvement.

Abbreviations: DRL, deep reinforcement learning; SLA, service level agreement.

```python
from flask import Flask, request, jsonify
import pandas as pd
import joblib
import time
import json

app = Flask(__name__)

# Load model
model = joblib.load('xgboost_model.pkl')

# Load classification report
with open('classification_report_xgboost.json', 'r') as f:
    metrics = json.load(f)

@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Get the input data
        data = request.get_json()

        # Ensure all required features are present
        required_features = [
            'age', 'BMI', 'heart rate', 'Respiratory rate', 'temperature',
            'SP O2', 'gendera_1', 'gendera_2', 'hypertensive_0', 'hypertensive_1',
            'diabetes_0', 'diabetes_1'
        ]
        if not all(feature in data for feature in required_features):
            return jsonify({'error': 'Missing required features. Please provide all required fields.'})

        # Create DataFrame from input data
        input_data = pd.DataFrame([data])

        # Start timer for prediction time
        start_time = time.time()

        # Make prediction
        prediction = model.predict(input_data)[0]

        # Calculate time taken for prediction
        prediction_time = time.time() - start_time

        # Prepare response
        response = {
            'prediction': int(prediction),
            'prediction_time': f"{prediction_time:.4f} seconds",
            'metrics': {
                'precision': metrics['1']['precision'],
                'recall': metrics['1']['recall'],
                'f1_score': metrics['1']['f1-score'],
                'accuracy': metrics['accuracy'],
                'support': metrics['1']['support']
            }
        }

        return jsonify(response)

    except Exception as e:
        return jsonify({'error': str(e)})

@app.route('/', methods=['GET'])
def home():
    return "Welcome to the prediction API! Use /predict endpoint for predictions."

if __name__ == '__main__':
    app.run(debug=True)
```

FIGURE 16 XGBoost model deployment via Flask.

stemming from the DT implementation through the creation of a digital prototype, as shown in Figure 18(b) and (c).

8.6 | Model performance analysis

Figure 19 shows the comparison of telemetry transmission latency for our model. It is observed that the latency increases steadily over time for both cases. However, the Pyomo model significantly optimised the latency, maintaining it around 20–25 milliseconds, while without Pyomo, the latency peaks at

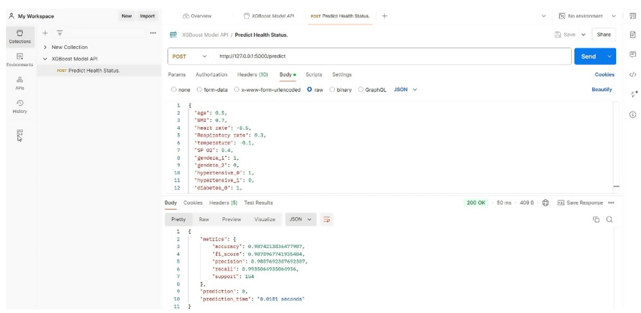
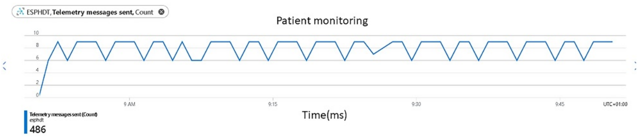


FIGURE 17 API deployment testing in Postman for real-time prediction. The Postman interface is used to send a request to the model deployed using Flask, showcasing the real-time prediction metrics. API, application programming interface.

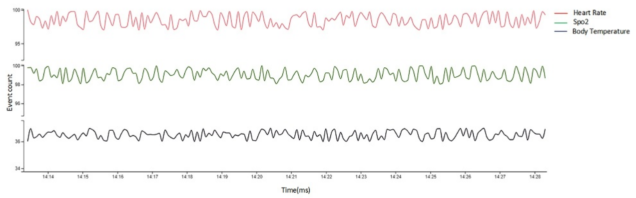
40 milliseconds. This improvement demonstrated that integrating the Pyomo model achieved a 32% reduction in telemetry transmission latency, leading to better real-time performance for healthcare monitoring..

Figure 20 presented three performance parameters: runtime (in seconds), data sent (in bytes), and CPU usage (in percentage). A noticeable trend is the improved CPU efficiency with our model, where CPU usage remains relatively stable compared to the fluctuating performance without Pyomo. Data transmission and runtime also exhibit improved consistency with the presented model. Overall, the integration of the Pyomo model results in an 18% improvement in resource utilisation, enhancing both data transmission efficiency and computational stability within the system.

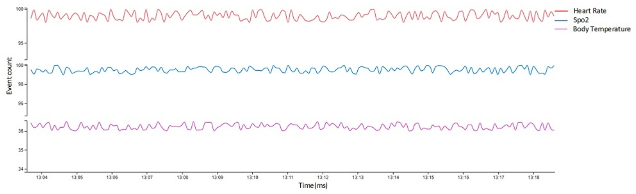
Figure 21 illustrated results indicating a significant reduction in response time when the proposed model is integrated into the system. The optimisation achieved through the presented model results in an approximate 52% reduction in response time, enhancing the system's efficiency in processing and responding to data.



(a) Data volume (Kbit) transmitted from the device to the cloud in real time.



(b) Data sent from a physical sensor to the cloud is visualised in TSI.



(c) Data sent from a virtual sensor to the cloud is visualised in TSI.

FIGURE 18 (a), (b) and (c) Data (heart rate (HR), Spo_2, and body temperature (BT)) monitoring accessing the cloud server using the dashboard on the cloud in TSI.

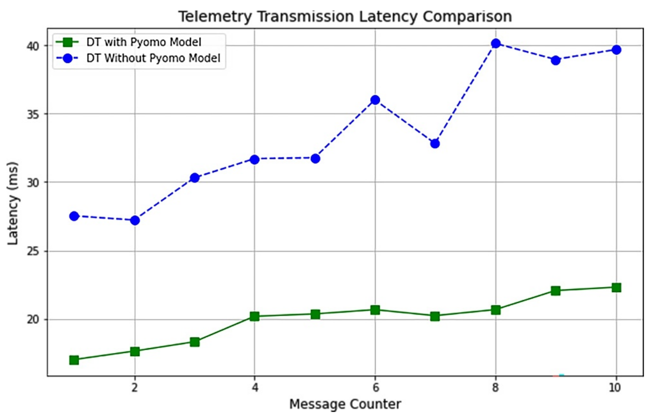


FIGURE 19 Comparison of transmission latency over time.

8.7 | Challenges and limitations of the proposed framework

To establish the DT environment, several challenges were encountered related to sensor integration, cloud communication, and system scalability. Specifically, during the data collection phase, we used the Arduino IDE to programme IoT devices, such as the ESP8266, which were connected to various sensors for monitoring health parameters like SpO2, HR, and BT. Configuring the hardware and ensuring real-time data transmission from the sensors to the cloud introduced challenges in terms of maintaining connectivity, especially in dynamic environments with varying network conditions.

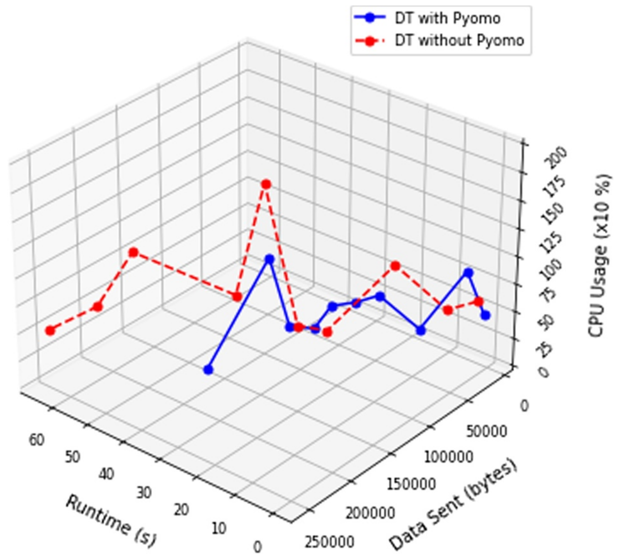


FIGURE 20 Performance comparison of runtime, data transmission, and CPU usage for digital twin (DT) systems with and without Pyomo.

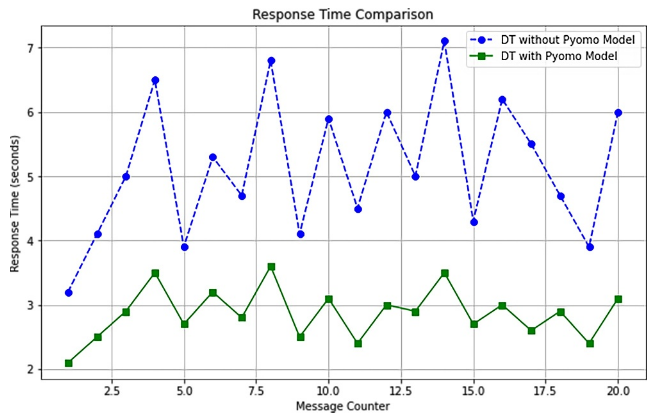


FIGURE 21 Response time comparison.

For cloud integration, the use of C# and JSON-LD for modelling DTs proved effective but presented complexity in data synchronisation and representation between physical and virtual models. Challenges were also faced in ensuring secure and efficient real-time data flow between sensors and the cloud, particularly when handling larger data volumes. Moreover, the system's reliance on external tools, such as the ADT explorer, introduced some limitations in the flexibility of queries and real-time visualisation capabilities. These factors necessitated additional customisations and optimisations to ensure efficient monitoring and analysis.

Despite these challenges, the framework was designed to be scalable and adaptable, capable of supporting real-time updates. However, the reliance on multiple software components could potentially limit its ease of integration with existing healthcare infrastructure without further customisation. Additionally, ensuring interoperability between different sensor types and managing the varying formats of the collected data posed some integration challenges, which were addressed through the use of standardised protocols like JSON-LD.

9 | CONCLUSION

This study successfully developed a system prototype utilising DT (DT) methodology, IoT, ML, and AI techniques to enhance data integration and interaction within healthcare. The system enables intelligent monitoring of physiological parameters, for example, HR, oxygen saturation (SpO2), and BT. The implementation of DT in healthcare has demonstrated significant potential in supporting cloud-based services for elderly individuals and those with chronic conditions. Integrated with a real-time graphical user interface based on ADT, the system allows both clinicians and patients to effectively manage and monitor health. The wearable prototype is lighter, smaller, and more cost-effective, facilitating the continuous monitoring of vital signs.

Edge computing methodologies have been employed to ensure prompt local assessments, reduce latency, and detect anomalies. The integration of the Pyomo model resulted in a 32% reduction in telemetry transmission latency, a 52% reduction in response time, and an 18% improvement in resource utilisation, demonstrating the system's optimisation effectiveness. Additionally, the portability and wireless nature of the device enhance usability. Machine learning techniques were employed to develop predictive models, achieving a 98% accuracy rate in real-time monitoring and 99.06% accuracy under cross-validation (cv = 20) using the XGBoost algorithm, which outperformed others with a training time of 0.25 s.

Future developments may involve incorporating additional sensors to monitor more vital signs, such as electrocardiograms, blood pressure, and GPS, to further refine the DT representation of patients. Moreover, the integration of 3D technology and advanced ML in healthcare applications holds considerable potential for further innovation in this domain.

AUTHOR CONTRIBUTIONS

Ahmed K. Jameil has made substantial contributions to the conceptualisation and design of the work; acquisition, analysis, and interpretation of data; developed the methodology; provided essential resources and software tools; validated the results and methods used; visualised the data; and played a primary role in writing the original draft of the manuscript.

Hamed Al Raweshidy has contributed to the funding acquisition, overseeing and leading the investigation process; managed project administration and coordination; supplied necessary resources; ensured the accuracy and validity of the procedures and results; supervised the project; and revised the manuscript critically for important intellectual content.

Both authors agree to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

ACKNOWLEDGEMENTS

Brunel University London.

CONFLICT OF INTEREST STATEMENT

The authors confirm that there is no conflict of interest related to this work.

DATA AVAILABILITY STATEMENT

Data is available on request from the authors.

ORCID

Ahmed K. Jameil https://orcid.org/0000-0002-1864-9807

REFERENCES

1. Kumar, P., Silambarasan, K.: Enhancing the performance of healthcare service in iot and cloud using optimized techniques. IETE J. Res. 68(2), 1475–1484 (2022). https://doi.org/10.1080/03772063.2019.1654934
2. Mazloomi, N., Gholipour, M., Zaretalab, A.: A priority-based congestion avoidance scheme for healthcare wireless sensor networks. IET Wirel. Sens. Syst. 13(1), 9–23 (2023). https://doi.org/10.1049/wss2.12046
3. Saleh, H., et al.: A scenario-based approach to the implementation of refueling stations in drone-based non-emergency of blood supply transportation. Arabian J. Sci. Eng. (2024). https://doi.org/10.1007/s13369-024-09549-7
4. Douaioui, K., Benmoussa, O., Ahlaqqach, M.: Optimizing supply chain efficiency using innovative goal programming and advanced metaheuristic techniques. Appl. Sci. 14(16), 7151 (2024). https://doi.org/10.3390/app14167151
5. KamelBoulos, M.N., Zhang, P.: Digital twins: from personalised medicine to precision public health. J. Personalized Med. 11(8), 745 (2021). https://doi.org/10.3390/jpm11080745
6. AlabdelAbass, A.A., Alshaheen, H., Takruri, H.: A game theoretic approach to wireless body area networks interference control. IET Wirel. Sens. Syst. (2024)
7. Hidalgo, A., Pérez, N., LemusAguilar, I.: Factors determining the success of ehealth innovation projects. Int. J. Software Sci. Comput. Intell. 14(1), 1–22 (2022). https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJSSCI.309709
8. Sun, W., et al.: Reducing offloading latency for digital twin edge networks in 6g. IEEE Trans. Veh. Technol. 69(10), 12240–12251 (2020). https://doi.org/10.1109/tvt.2020.3018817
9. Elayan, H., Aloqaily, M., Guizani, M.: Digital twin for intelligent context-aware iot healthcare systems. IEEE Internet Things J. 8(23), 16749–16757 (2021). https://doi.org/10.1109/jiot.2021.3051158
10. De.Benedictis, A., et al.: Digital twins in healthcare: an architectural proposal and its application in a social distancing case study. IEEE J. Biomed. Health Inf. (2022)
11. Punn, N.S., et al.: Monitoring Covid-19 Social Distancing with Person Detection and Tracking via Fine-Tuned Yolo v3 and Deepsort Techniques (2020). arXiv preprint arXiv:200501385
12. Schroeder, G.N., et al.: Digital twin data modeling with automationML and a communication methodology for data exchange. IFAC-PapersOnLine 49(30), 12–17 (2016). https://doi.org/10.1016/j.ifacol.2016.11.115
13. Hassani, H., Huang, X., MacFeely, S.: Impactful digital twin in the healthcare revolution. Big Data Cogn. Comput. 6(3), 83 (2022). https://doi.org/10.3390/bdcc6030083
14. Mhamdi, L., AbdulKhalek, H.: Congestion control in constrained internet of things networks. IET Wirel. Sens. Syst. 13(6), 247–255 (2023). https://doi.org/10.1049/wss2.12072
15. Jayaraju, N., et al.: Mobile phone and base stations radiation and its effects on human health and environment: a review. Sustain. Technol. Entrepreneurship 2(2), 100031 (2023). https://doi.org/10.1016/j.stae.2022.100031
16. Thamotharan, P., et al.: Human digital twin for personalized elderly type 2 diabetes management. J. Clin. Med. 12(6), 2094 (2023). https://doi.org/10.3390/jcm12062094
17. Wang, B., et al.: Human digital twin in the context of industry 5.0. Robot. Comput. Integrated Manuf. 85, 102626 (2024). https://doi.org/10.1016/j.rcim.2023.102626
18. Ma, X., Qi, Q., Tao, F.: An ontology-based data-model coupling approach for digital twin. Robot. Comput. Integrated Manuf. 86, 102649 (2024). https://doi.org/10.1016/j.rcim.2023.102649
19. Fang, K., et al.: Idres: Identity-based respiration monitoring system for digital twins enabled healthcare. IEEE J. Sel. Area. Commun. 41(10), 3333–3348 (2023). https://doi.org/10.1109/jsac.2023.3310095

20. Ren, K., Byun, Y., Wilder, B.: Decision-focused Evaluation of Worst-Case Distribution Shift (2024). arXiv preprint arXiv:240703557
21. Mostajabdaveh, M., et al.: Optimization modeling and verification from problem specifications using a multi-agent multi-stage llm framework. INFOR Inf. Syst. Oper. Res. 62(4), 1–19 (2024). https://www.tandfonline.com/doi/full/10.1080/03155986.2024.2381306
22. Khan, S., et al.: A novel digital twin (dt) model based on wifi csi, signal processing and machine learning for patient respiration monitoring and decision-support. IEEE Access 11, 103554–103568 (2023). https://doi.org/10.1109/access.2023.3316508
23. Su, J., et al.: A real-time cross-domain wi-fi-based gesture recognition system for digital twins. IEEE J. Sel. Area. Commun. 41(11), 3690–3701 (2023). https://doi.org/10.1109/jsac.2023.3310073
24. Shankar, K., et al.: Synergic deep learning for smart health diagnosis of Covid-19 for connected living and smart cities. ACM Trans. Internet Technol. 22(3), 1–14 (2022). https://doi.org/10.1145/3453168
25. Abilkaiyrkyzy, A., et al.: Dialogue system for early mental illness detection: toward a digital twin solution. IEEE Access 12, 2007–2024 (2024). https://doi.org/10.1109/access.2023.3348783
26. Maimour, M., Ahmed, A., Rondeau, E.: Survey on digital twins for natural environments: a communication network perspective. Internet of Things 25, 101070 (2024). https://doi.org/10.1016/j.iot.2024.101070
27. Chen, J., et al.: Digital twin empowered wireless healthcare monitoring for smart home. IEEE J. Sel. Area. Commun. 41(11), 3662–3676 (2023). https://doi.org/10.1109/jsac.2023.3310097
28. Alqahtani, A., Alsubai, S., Bhatia, M.: Digital twin-assisted healthcare framework for adult. IEEE Internet Things J. 11(8), 1–14970 (2023). https://doi.org/10.1109/jiot.2023.3345331
29. Mohamed, N., et al.: Leveraging digital twins for healthcare systems engineering. IEEE Access 11, 69841–69853 (2023). https://doi.org/10.1109/access.2023.3292119
30. Avanzato, R., et al.: Lung-dt: an ai-powered digital twin framework for thoracic health monitoring and diagnosis. Sensors 24(3), 958 (2024). https://doi.org/10.3390/s24030958
31. Li, X., et al.: Human-centric manufacturing for human-system coevolution in industry 5.0. CIRP Annals 72(1), 393–396 (2023). https://doi.org/10.1016/j.cirp.2023.04.039
32. Viceconti, M., et al.: Position paper from the digital twins in healthcare to the virtual human twin: a moon-shot project for digital health research. IEEE J. Biomed. Health Inf. 28(1), 491–501 (2024). https://doi.org/10.1109/jbhi.2023.3323688
33. Grieves, M.: Digital Twin: Manufacturing Excellence through Virtual Factory Replication - A Whitepaper by Dr. Michael Grieves'(March), 1–7 (2014). White Paper
34. Tao, F., et al.: Digital twin driven prognostics and health management for complex equipment. CIRP Annals 67(1), 169–172 (2018). https://doi.org/10.1016/j.cirp.2018.04.055
35. Coorey, G., et al.: The health digital twin to tackle cardiovascular disease—a review of an emerging interdisciplinary field. NPJ Digit. Med. 5(1), 126 (2022). https://doi.org/10.1038/s41746-022-00640-7
36. Yang, Q., et al.: Development of digital fetal heart models with virtual ultrasound function based on cardiovascular casting and computed tomography scan. Bioengineering 9(10), 524 (2022). https://doi.org/10.3390/bioengineering9100524
37. Gillette, K., et al.: A framework for the generation of digital twins of cardiac electrophysiology from clinical 12-leads ecgs. Med. Image Anal. 71, 102080 (2021). https://doi.org/10.1016/j.media.2021.102080
38. NaitAbbou, A., Manner, J.: Etxre: energy and delay efficient routing metric for rpl protocol and wireless sensor networks. IET Wirel. Sens. Syst. 13(6), 235–246 (2023). https://doi.org/10.1049/wss2.12070
39. Wang, W., et al.: Digital twin rehabilitation system based on self-balancing lower limb exoskeleton. Technol. Health Care 31(1), 103–115 (2023). https://doi.org/10.3233/thc-220087
40. Lv, Z., Guo, J., Lv, H.: Deep learning-empowered clinical big data analytics in healthcare digital twins. IEEE ACM Trans. Comput. Biol. Bioinf 21(4), 1–11 (2023). https://doi.org/10.1109/tcbb.2023.3252668
41. Liu, Y., et al.: A novel cloud-based framework for the elderly healthcare services using digital twin. IEEE Access 7, 49088–49101 (2019). https://doi.org/10.1109/access.2019.2909828
42. Lin, Y., et al.: A novel architecture combining oracle with decentralized learning for iiot. IEEE Internet Things J. 10(5), 3774–3785 (2023). https://doi.org/10.1109/jiot.2022.3150789
43. Lutze, R.: Digital twins in ehealth–: prospects and challenges focussing on information management. In: 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), pp. 1–9. IEEE (2019)
44. Sahal, R., Alsamhi, S.H., Brown, K.N.: Personal digital twin: a close look into the present and a step towards the future of personalised healthcare industry. Sensors 22(15), 5918 (2022). https://doi.org/10.3390/s22155918
45. Quinn, A.R., et al.: A digital twin framework for robust control of robotic-biological systems. J. Biomech. 152, 111557 (2023). https://doi.org/10.3233/thc-220087
46. Voigt, I., et al.: Digital twins for multiple sclerosis. Front. Immunol. 12, 12 (2021). https://doi.org/10.3389/fimmu.2021.669811
47. Hajar, M.S., Al.Kadri, M.O., Kalutarage, H.K.: A survey on wireless body area networks: architecture, security challenges and research opportunities. Comput. Secur. 104, 102211 (2021). https://doi.org/10.1016/j.cose.2021.102211
48. Chen, M., et al.: Living with i-fabric: smart living powered by intelligent fabric and deep analytics. IEEE Network 34(5), 156–163 (2020). https://doi.org/10.1109/mnet.011.1900570
49. Yu, F., et al.: Smart clothing system with multiple sensors based on digital twin technology. IEEE Internet Things J. 10(7), 6377–6387 (2022). https://doi.org/10.1109/jiot.2022.3224947
50. KraneGartiser, K., et al.: Actigraphy as an objective intra-individual marker of activity patterns in acute-phase bipolar disorder: a case series. Int. J. Behav. Dev. 6(1), 1–7 (2018)
51. FernandezRuiz, I.: Computer modelling to personalize bioengineered heart valves. Nat. Rev. Cardiol. 15(8), 440–441 (2018). https://doi.org/10.1038/s41569-018-0040-x
52. Laaki, H., Miche, Y., Tammi, K.: Prototyping a digital twin for real time remote control over mobile networks: application of remote surgery. IEEE Access 7, 20325–20336 (2019). https://doi.org/10.1109/access.2019.2897018
53. MartinezVelazquez, R., Gamez, R., El.Saddik, A.: Cardio twin: a digital twin of the human heart running on the edge. In: 2019 IEEE International Symposium on Medical Measurements and Applications (MeMeA), pp. 1–6 (2019)
54. Jameil, A.K., Al.Raweshidy, H.: Efficient cnn architecture on fpga using high level module for healthcare devices. IEEE Access 10, 60486–60495 (2022). https://doi.org/10.1109/access.2022.3180829
55. Zhong, Y., et al.: Idtvr: a novel cloud framework for an interactive digital twin in virtual reality. In: 2022 IEEE 2nd International Conference on Intelligent Reality (ICIR), pp. 21–26. IEEE (2022)
56. Jia, P., Wang, X., Shen, X.: Accurate and efficient digital twin construction using concurrent end-to-end synchronization and multi-attribute data resampling. IEEE Internet Things J. 10(6), 4857–4870 (2022). https://doi.org/10.1109/jiot.2022.3221012
57. Drummond, D., Roukema, J., Pijnenburg, M.: Home monitoring in asthma: towards digital twins. Curr. Opin. Pulm. Med. 29(4), 270–276 (2023). Publish Ahead of Print. https://doi.org/10.1097/mcp.0000000000000963
58. Drummond, D.: Digital tools for remote monitoring of asthma patients: gadgets or revolution? Rev. Mal. Respir. 39(3), 241–257 (2022). https://doi.org/10.1016/j.rmr.2022.01.018
59. Islam, M.M., Rahaman, A., Islam, M.R.: Development of smart healthcare monitoring system in iot environment. SN Comput. Sci. 1(3), 185 (2020). https://doi.org/10.1007/s42979-020-00195-y
60. Xie, Y., et al.: Integration of artificial intelligence, blockchain, and wearable technology for chronic disease management: a new paradigm in smart healthcare. Curr. Med. Sci. 41(6), 1123–1133 (2021). https://doi.org/10.1007/s11596-021-2485-0

61. Zhang, L., Xu, X.: Corrigendum: construction of smart older adults care service model driven by primary health care. Front. Public Health 11, 1319932 (2023). https://doi.org/10.3389/fpubh.2023.1319932
62. Mohamed, N., et al.: How healthcare systems engineering can benefit from digital twins? In: 2023 IEEE International Systems Conference (SysCon), pp. 1–6. IEEE, Vancouver (2023). https://ieeexplore.ieee.org/document/10131101/
63. Nie, Q., et al.: A multi-agent and cloud-edge orchestration framework of digital twin for distributed production control. Robot. Comput. Integrated Manuf. 82, 102543 (2023). https://doi.org/10.1016/j.rcim.2023.102543
64. Qu, Z., et al.: Iomt-based smart healthcare detection system driven by quantum blockchain and quantum neural network. IEEE J. Biomed. Health Inf. 28(6), 3317–3328 (2024). https://doi.org/10.1109/jbhi.2023.3288199
65. Alqahtani, A., Alsubai, S., Bhatia, M.: Digital-twin-assisted healthcare framework for adult. IEEE Internet Things J. 11(8), 14963–14970 (2024). https://doi.org/10.1109/jiot.2023.3345331
66. INTEGRATED, M.: Max30102 high-sensitivity pulse oximeter and heart-rate sensor for wearable health. Datasheet, Rev 1, 32 (2022)
67. Quirita, P.E.A., Troncoso, L.J.: Coefficients search for a regression curve of a neonatal incubator using evolutionary computing techniques. In: 2022 IEEE XXIX International Conference on Electronics, Electrical Engineering and Computing (INTERCON), pp. 1–4. IEEE (2022)
68. Li, F., et al.: Prediction model of in-hospital mortality in intensive care unit patients with heart failure: machine learning-based, retrospective analysis of the MIMIC-III database. BMJ Open 11(7), 1–17 (2021). https://doi.org/10.1136/bmjopen-2020-044779
69. Erdoğan, B., Oğul, H.: Objective pain assessment using vital signs. Proc. Comput. Sci. 170, 947–952 (2020). https://doi.org/10.1016/j.procs.2020.03.103
70. Minnesota Department of Health: Oxygen levels. Pulse Oximeters, and COVID-19 19, 1–2 (2021)
71. Zhang, D., Gong, Y.: The comparison of lightgbm and xgboost coupling factor analysis and prediagnosis of acute liver failure. IEEE Access 8, 220990–221003 (2020). https://doi.org/10.1109/access.2020.3042848
72. ShalevShwartz, S., BenDavid, S.: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press (2014)
73. Jameil, A.K., Al-Raweshidy, H.: Enhancing offloading with cybersecurity in edge computing for digital twin-driven patient monitoring. IET Wirel. Sens. Syst. (2024). p. wss2.12086. https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/wss2.12086
74. Hassan, M., Kelsey, T., Khan, B.M.: Elderly care and health monitoring using smart healthcare technology: an improved routing scheme for wireless body area networks. IET Wirel. Sens. Syst. (2024). p. wss2.12097. https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/wss2.12097
75. HosseinShokouhi, M., Hadi, M., Pakravan, M.R.: Mobility-aware computation offloading for hierarchical mobile edge computing. IEEE Trans. Network Serv. Manag. 21(3), 3372–3384 (2024). https://doi.org/10.1109/tnsm.2024.3386845
76. Wang, J., et al.: Multi-agent reinforcement learning for task offloading with hybrid decision space in multi-access edge computing. Ad Hoc Netw. 166, 103671 (2025). https://doi.org/10.1016/j.adhoc.2024.103671
77. Lan, S., et al.: Sla-orecs: an sla-oriented framework for reallocating resources in edge-cloud systems. J. Cloud Comput. 13(1), 18 (2024). https://doi.org/10.1186/s13677-023-00561-0
78. EmamiKhansari, M., Sharifian, S.: A deep reinforcement learning approach towards distributed function as a service (faas) based edge application orchestration in cloud-edge continuum. J. Netw. Comput. Appl. 233, 104042 (2025). https://doi.org/10.1016/j.jnca.2024.104042
79. Wang, Y., et al.: Efficient task migration and resource allocation in cloud–edge collaboration: a drl approach with learnable masking. Alex. Eng. J. 111, 107–122 (2025). https://doi.org/10.1016/j.aej.2024.10.015
80. Min, H., et al.: A joint optimization of resource allocation management and multi-task offloading in high-mobility vehicular multi-access edge computing networks. Ad Hoc Netw. 166, 103656 (2025). https://doi.org/10.1016/j.adhoc.2024.103656

How to cite this article: Jameil, A.K., Al-Raweshidy, H.: Implementation and evaluation of digital twin framework for Internet of Things based healthcare systems. IET Wirel. Sens. Syst. 14(6), 507–527 (2024). https://doi.org/10.1049/wss2.12101