High-Performance Hybrid AI Systems with Quantum-Secure Protocols for Cyber-Physical Remote Healthcare Applications

By

Ahmed K. Jameil

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy (Ph.D.)

of the

Department of Electronic and Electrical Engineering,

College of Engineering, Design and Physical Sciences

Brunel University London

July 2025

To the most wonderful person in the world, my father's soul...
To everyone who supported me with unwavering love and sincerity...
To my beloved family and cherished friends...
Thank you for your endless inspiration, love, and support.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and has not been submitted, in whole or in part, for any other degree or qualification at this or any other institution. This thesis represents the results of my own research, analysis, and writing. I have ensured that all information and data derived from other sources have been appropriately cited and acknowledged.

Ahmed K. Jameil
July 2025

Acknowledgments

I would like to extend my heartfelt gratitude to the Ministry of Higher Education and Scientific Research, the Cultural Attaché, and the **University of Diyala** in Iraq for their generous funding and invaluable cooperation in facilitating my Ph.D. studies.

I am also deeply thankful to **Professor Hamed Al-Raweshidy** for his insightful advice, guidance, and unwavering support. His firm belief in my potential, enthusiastic encouragement, and our constructive dialogues have been immensely valuable. What I appreciate most is his patience and understanding, as he allowed me the freedom to explore my own areas of study. His constant encouragement to publish in high-impact journals has greatly contributed to my growth and confidence in the field of scientific research. Professor Hamed, I am forever indebted to you and will always remember your kindness and support.

I would also like to express my appreciation to the Student Center at Brunel University London and extend my gratitude to all the staff and my colleagues at Brunel University London for their support.

Moreover, I deeply value the crucial role my family has played in providing unwavering support, love, inspiration, and encouragement throughout my academic journey.

To my dear daughter **Fatima**, I cannot turn back time, but I will strive to make up for the moments we missed together.

Finally, I want to honour my loving, supportive and patient **Father: Khudaer Jameil Hamad**. He once said that I might go to study in the UK, but by the time I finish, he might not be here to see it. His words, sadly, came true.

Abstract

Digital Twin (DT) technology is increasingly important for real-time healthcare monitoring and predictive analytics. However, existing healthcare systems face critical challenges, including excessive computational load, high network latency, vulnerability to quantum cyberattacks, and inefficient strategies for distributing tasks across cloud and edge environments. Existing solutions often fail to scale efficiently, protect sensitive health data against future cyberattacks, or deliver reliable performance under dynamic conditions. To address these challenges, this work proposes an integrated healthcare framework that advances the state-of-the-art across multiple dimensions.

First, to tackle the computational bottleneck in wearable healthcare devices, a lightweight one-dimensional convolutional neural network (CNN) accelerator was designed and implemented on field-programmable gate arrays (FPGAs), leveraging shift-based computation and pipelined architecture. This achieved a classification throughput of 1145 GOPS (Giga operations per second), enabling ultra-low-latency and energy-efficient biosignal analysis. Second, to enhance system responsiveness and scalability, a cloud-edge Digital Twin healthcare system was developed, leveraging secure Internet of Things (IoT) communication, dynamic telemetry optimization using Pyomo mathematical programming, and real-time predictive analytics. Third, to address emerging data security threats, a novel quantum-secure healthcare Digital Twin model was introduced, leveraging Quantum Key Distribution (QKD) protocols and hybrid artificial intelligence (AI) models combining multilayer perceptrons (MLP), extreme gradient boosting (XGBoost), and generative adversarial networks (GANs) for data augmentation. Finally, to optimize system resilience under dynamic healthcare conditions, a dynamic task offloading strategy was proposed, leveraging multi-agent reinforcement learning (MAPPO), adaptive cybersecurity protection (ACTO), and quantum-enhanced task preprocessing (AQDT-IoT).

Experimental results demonstrate that the FPGA accelerator achieves 1145 GOPS throughput for real-time biosignal classification, while the proposed cloud-edge healthcare system reduces network latency by 40% and improves throughput by 30%. The hybrid AI model achieves an average prediction accuracy of 97.48% across health indicators under 10-fold cross-validation. Moreover, the adaptive task offloading framework increases task success rates by 32% and reduces error rates by 80%, significantly improving operational efficiency and system robustness. Compared to previous approaches, the proposed framework delivers a highly scalable, secure, and intelligent Digital Twin Healthcare system, significantly strengthening patient monitoring, predictive decision-making, and preparedness against future quantum-era cybersecurity threats.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

1-D CNN	One-Dimensional Convolutional Neural Network
A3C	Asynchronous Advantage Actor-Critic
ACTO	Adaptive Cybersecurity Task Offloading
AI	Artificial Intelligence
API	Application Programming Interface
AUC	Area Under the Curve
AXI bus	Advanced eXtensible Interface bus
BN	Bayesian Networks
BT	Body Temperature
CIA	Confidentiality, Integrity, and Availability
CNN	Convolutional Neural Network
COPD	Chronic Obstructive Pulmonary Disease
CPU	Central Processing Unit
CPS	Cyber-Physical Systems
D2D	Device-to-Device Communication
D2C	Device-to-Cloud Communication
DQN	Deep Q-Network
DSPs	Digital Signal Processors
DT	Digital Twin
DTE	Digital Twin Environment
DTDL	Digital Twin Definition Language
DTEN	Digital Twin Enabled Networks
DTH	Digital Twin Healthcare
DTNMHM	Digital Twins Network Model for Healthcare Monitoring
ECG	Electrocardiogram
EE	Energy Efficiency
EEG	Electroencephalogram
EMG	Electromyography
EHR	Electronic Health Record
ER	Error Rate
FC	Fully Connected
FSM	Finite State Machine

FPGA	Field-Programmable Gate Array
FFs	Flip-Flops
GHz	Gigahertz
GAN	Generative Adversarial Network
GLB	Generic Logic Block
GMM	Gaussian Mixture Modeling
GOPS	Giga Operations Per Second
GPU	Graphics Processing Unit
HIPAA	Health Insurance Portability and Accountability Act
HR	Heart Rate
HIoT	Healthcare Internet of Things
HMS	Healthcare Monitoring System
IDEs	Integrated Development Environments
IoT	Internet of Things
ISC	Initiate Secure Channel
KNN	K-Nearest Neighbor
LEDs	Light-Emitting Diodes
LR	Logistic Regression
LTI	Linear Time-Invariant
LUTs	Look-Up Tables
MAPPO	Multi-Agent Proximal Policy Optimization
MES	Mobile Edge Servers
MEC	Mobile Edge Computing
MHz	Megahertz
ML	Machine Learning
MLP	Multilayer Perceptron
MME	Matrix Multiplication Engine
MQTT	Message Queuing Telemetry Transport
NB	Naive Bayes
NL	Network Latency
NMF	Non-Negative Matrix Factorization
NN	Neural Network
NoC	Network-on-Chip
OWC	Optical Wireless Communication
PaaS	Platform as a Service
PCA	Principal Component Analysis
PE	Processing Element
PHS	Personal Health Systems
PoC	Proof of Concept

QKD	Quantum Key Distribution
RBAC	Role-Based Access Control
ReLU	Rectified Linear Unit
RSA	Rivest–Shamir–Adleman Cryptosystem
RTP	Real-Time Pattern
SGD	Stochastic Gradient Descent
SpO2	Oxygen Saturation
SVM	Support Vector Machine
SR	System Responsiveness
TSI	Time Series Insights
V2I	Vehicle-to-Infrastructure Communication
V2V	Vehicle-to-Vehicle Communication
V2X	Vehicle-to-Everything Communication
WBANs	Wireless Body Area Networks
XAI	Explainable Artificial Intelligence
XGBoost	eXtreme Gradient Boosting Algorithm
T	Temperature
TOSR	Task Offloading Success Rate
H	Humidity
HIS	Hospital Information System

List of Symbols

$\mathcal{S}_D$	Patient status
$\mathcal{F}_{DT}$	Digital Twin model with Machine Learning
D_ξ	Supplementary data flow rate
$\mathcal{H}_D$	Historical health data
$\mathcal{Z}_t$	Normalized data at time t
$\mathfrak{I}_{df}$	Data flow rate
$\mathcal{S}_{policy}$	Shared access policy
$\mathcal{S}_t(t)$	Measured database at time t
$\mathcal{P}_{data}$	Patient data
$\mathcal{P}_{policy}$	Policy details
$\mathcal{F}_m$	Forecasting model of operational process
$C_\mathcal{L}$	Communication layer
$C_{\mathcal{L}-PD}$	Communication between physical objects and DT data
$C_{\mathcal{L}-VD}$	Communication between virtual objects and DT data
$C_{\mathcal{L}-PV}$	Communication between physical and virtual objects
$\mathfrak{D}_{DT}$	Digital Twin data
s_t	Current state
s_{t+1}	Next state
a_t	Action selected by the ML model
R_t	Reward at time t
$L(\theta)$	Loss function
$\hat{R}_t(\theta)$	Predicted reward based on model parameters
X_t	Extracted feature vector
$\frac{\partial L}{\partial \theta}$	Gradient of the loss function with respect to model parameters
η	Learning rate
$\mathcal{P}\rangle$	Personal information
$\mathcal{S}\Updownarrow$	Medical sensor data
$\mathcal{B}\Updownarrow$	Behavioral model
$\mathcal{N}\rfloor$	Network connection state
$\mathcal{E}$	Event grid data
$\mathcal{DTC}$	Digital Twin Client
$\mathcal{DM}$	Device Message
$\mathcal{DI}$	Device ID

$\mathcal{DS}$	Device sensor data
$\mathcal{UD}$	Update Digital Twin data
$\mathcal{M}_{DT}$	Digital Twin model data
$\mathcal{DC}(t)$	Data Collection function at time t
$\mathcal{B}_{\text{IoT}}(t)$	IoT data latency management function
$\mathcal{R}_{DT}$	Data transmission rate
Θ_{trans}	Transmission time
S_{required}	Required storage size
D_v	Average data volume per patient
$\mathcal{E}(t)$	Stored and predicted patient data
Θ_{DEHR}	Secure data transmission to EHR
R_{EHR}	Response from Electronic Health Records system
R_{HIS}	Response from Hospital Information System
M_{data}	Mapping data container
Λ_{DMHIS}	Mapping of hospital data to digital format
Ξ	Scaled system resources
Θ	Threshold parameter
ρ	Resource allocation factor
ΔD	Data volume change
Υ	Backup service status
Ω	Balanced data processing
Λ_{LBD}	Load balancing function
Σ_{RFT}	Redundancy and fault tolerance factor
Φ_{DCC}	Dynamic cloud capacity
$\Upsilon_{\text{high}}, \Upsilon_{\text{low}}$	Upper and lower system thresholds
Θ_{DR}	Disaster recovery trigger
$\mathcal{E}_K(D)$	Encryption of data D
C	Ciphertext
$\mathcal{H}(D)$	Hash function applied to D
H'	Computed hash value
K_{pubR}	Receiver's public key
K_{priS}	Sender's private key
E	Encrypted data
S	Data signature
C	Secure communication channel
V	Signature validation flag
D_{dec}	Decrypted data
S_{recv}	Received signature
H_{recv}	Hash of decrypted data

N_i	Offloading necessity of task i
O_i	Offloading decision for task i
P_i	Partial offloading fraction
N_i'	Adjusted offloading necessity
$P_{\text{attack},i}$	Probability of mobile edge server i being attacked
S_i	Security score of mobile edge server i
T_i	Task offloading operation
H_i	Healthcare task
S_{net}	Current network status
θ	Offloading necessity threshold
N_{max}	Maximum task necessity across all tasks
A_{dt}	Accuracy of Digital Twin predictions
S_f	Social factors impact
α_1, α_2	Coefficients for biosignal effects (HR, BT)
$L(t)$	Light intensity at time t
$S(t)$	Sleepiness level at time t
ϵ_1	Light exposure impact factor
$R_1(t)$	Respiratory comfort at time t
ζ_1, ζ_2	Humidity and temperature impacts
$D(t)$	Device functionality at time t
η_1	Magnetic field impact factor
$R(t)$	Reliability assessment
$\lambda_1, \lambda_2, \lambda_3$	Weighting factors for metrics
$\omega_1, \omega_2, \omega_3$	Metric importance weights
P_{overall}	Overall task offloading performance
$E_{\text{consumption}}$	Energy consumption
T_{delay}	Task delay time
$H_{\text{efficiency}}$	Healthcare system efficiency
E_i	Energy consumed by task i
$T_{\text{transmission}}$	Transmission delay
$T_{\text{processing}}$	Processing delay

Publications

A. Papers Published: The following papers have been published:

[$\mathbf{J_{p1}}$] A. K. Jameil and H. Al-Raweshidy, "Efficient CNN Architecture on FPGA Using High-Level Module for Healthcare Devices," *IEEE Access*, vol. 10, pp. 60486–60495, 2022. doi: 10.1109/ACCESS.2022.3180829 [1].

[$\mathbf{J_{p2}}$] A. K. Jameil and H. Al-Raweshidy, "Enhancing Offloading with Cybersecurity in Edge Computing for Digital Twin-Driven Patient Monitoring," *IET Wireless Sensor Systems*, July 2024. doi: 10.1049/wss2.12086 [2].

[$\mathbf{J_{p3}}$] A. K. Jameil and H. Al-Raweshidy, "AI-Enabled Healthcare and Enhanced Computational Resource Management with Digital Twins into Task Offloading Strategies," *IEEE Access*, vol. 12, pp. 90353–90370, 2024. doi: 10.1109/ACCESS.2024.3420741 [3].

[$\mathbf{J_{p4}}$] A. K. Jameil and H. Al-Raweshidy, "Implementation and Evaluation of Digital Twin Framework for Internet of Things-Based Healthcare Systems," *IET Wireless Sensor Systems*, vol. 14, no. 6, pp. 507–527, 2024. https://doi.org/10.1049/wss2.12101 [4].

[$\mathbf{J_{p5}}$] A. K. Jameil and H. Al-Raweshidy, "A Digital Twin Framework for Real-Time Healthcare Monitoring: Leveraging AI and Secure Systems for Enhanced Patient Outcomes," *Discover Internet of Things*, vol. 5, p. 37, 2025. https://doi.org/10.1007/s43926-025-00135-3 [5].

[$\mathbf{J_{p6}}$] A. K. Jameil and H. Al-Raweshidy, "Quantum-Enhanced Digital Twin IoT for Efficient Healthcare Task Offloading," *Discover Applied Sciences*, vol. 7, p. 525, 2025. https://doi.org/10.1007/s42452-025-07101-2 [6].

[$\mathbf{J_{p7}}$] A. K. Jameil and H. Al-Raweshidy, "Hybrid Cloud-Edge AI Framework for Real-Time Predictive Analytics in Digital Twin Healthcare Systems," *Scientific Reports*, 2025. https://doi.org/10.21203/rs.3.rs-5412158/v1 [7].

B. Papers Submitted to Journals: The following papers have been submitted to journals:

[$\mathbf{J_{s8}}$] A. K. Jameil and H. Al-Raweshidy, "Quantum-Resistant Security in Digital Twin Healthcare Systems," *IET Wireless Sensor Systems*, submitted.

[$\mathbf{J_{s9}}$] A. K. Jameil and H. Al-Raweshidy, "Towards Unified Digital Twin Frameworks in Healthcare: Addressing Scalability, Security, and Interoperability Challenges ," *IEEE Internet of Things Journal*, submitted.

Notation:

$\mathbf{J_p}$: Journal paper published.

$\mathbf{J_s}$: Journal paper submitted.

Chapter 1

Introduction

1.1 Background and Context

Health monitoring devices have displaced fundamental medical platforms by ushering in a new era of tracking, analysing, and reacting to patient's raw health data flows. They are universally recognised as being at the frontier of healthcare advancement and remain to respond to key healthcare challenges, as well as shape the future state of health care. This also encompasses not only diagnosing diseases but also assessing patients outside the hospital and providing highly individualised approaches [8], [9].

Furthermore, FPGAs' use in healthcare changed the process of creating medical devices fundamentally in the sense of utilisation. Originally designed for incredibly efficient calculations, those devices became critical for the analysis of vital signs, temperature, blood pressure and any other parameters which require prompt detection in emergency situations. FPGA applications have been incorporated into various forms of monitoring systems such as ECG [10], EEG [11], and EMG [12] to break down detailed cardiac, neural, and muscular activity.

In addition, Convolutional Neural Networks (CNNs) have almost single handedly brought a revolutionary change in medical diagnosis. CNNs, trained on large amount of data, improve their diagnostic performance over time, adding the level of sophistication in diagnostics, unparalleled to any human work. Therefore, CNNs have been considered an unavoidable component in the CAD systems and have expanded the field of personalised medicine profoundly [13]. Powerful health monitoring systems have emerged from the synergy between FPGA and CNN technologies. The combination of computational resources and adaptability inherent to FPGAs with the deep analytical power of CNNs has resulted in highly innovative solutions that break new ground in patient care and monitoring. These advances illustrate the importance of computational technologies in healthcare and underscore continued efforts to pursue research and development to enhance patient outcomes and transform healthcare provision [14]. Historically, the development of computational methods has played an important role in research involving health monitoring technologies, largely influenced by advancements in FPGA and CNN technologies. As these technologies continue to evolve and integrate, the potential for further innovation in healthcare monitoring remains vast, promising a future where real-time, accurate, and personalized patient care is not just a possibility, but a reality [15], [16].

The digital transformation process currently transforming many industries, including healthcare, began with the launch of the industry 4.0 project in 2013. This approach relies heavily on advanced technologies, such as the Internet of Things (IoT), cloud and edge computing, and big data analytics. The digital twin (DT) paradigm, based on these technologies, allows for the digital transformation of any system and is commonly utilized by industrial and engineering companies. Over the past decade, DT technology has been recommended for healthcare applications, resulting in the development of Digital Twins Healthcare (DTH) [17].

Research in DT further encompasses the enhancement of wireless body area networks (WBANs), alongside the deployment of advanced signal processing and sensors that support DT, with the integration of Markov decision processes (MDP) and advanced computational techniques to elevate both efficiency and reliability in health monitoring systems [18]–[21]. Within smart homes, DT applications strive to improve the monitoring, prediction, and control of health parameters, utilising an array of wireless and wearable technologies. In the healthcare sector, professionals utilize DT in conjunction with cloud and IoT-edge computing, and sophisticated data analysis methods, aiming to deliver intelligent predictive diagnostics and secure health data management. This is all facilitated by the DT framework for assessments, significantly advancing monitoring capabilities [22].

The Lung-DT framework features a modular microservices architecture, integrating IoT sensor data and historical radiology for lung monitoring. It uses advanced preprocessing pipelines and YOLOv8 neural networks for accurate disease classification. Real-time simulation and predictive analytics are enabled through a dynamic digital twin. Designed for scalability, it employs Docker and Kubernetes. However, the lack of integrated data storage limits seamless patient history tracking and full accessibility for end-users [23]. Additionally, the virtual human twin, a detailed digital model of human pathophysiology, has been proposed to establish a collaborative infrastructure, which plays a crucial role in accelerating both the development and adoption of DT technologies in healthcare [24].

The intersection of advancements in FPGAs, CNNs, and DT technologies has set the stage for transformative shifts within the healthcare industry, enabling significant innovation (e.g., enhanced diagnostic precision and real-time monitoring systems). Together, they improve the accuracy and speed of patient care while making it more personalized, creating new possibilities as well as challenges in achieving better healthcare outcomes with greater operational efficiencies.

1.2 Significance of FPGA in Healthcare Monitoring

Amid the rapidly changing healthcare technology, FPGAs have served as key accelerators in healthcare monitoring systems. Their unique properties and flexibility give them unprecedented advantages over conventional computational approaches, enabling significantly more efficient and comprehensive patient care as well as an enhanced analytics experience [17].

1.2.1 Role and Advantages of FPGA

In the universe of healthcare technologies, FPGAs are a very special occurrence as they can be reconfigured and have tremendous processing speeds. FPGAs are nonstandard products (nonconventional processors) unlike normal flexible, inflexible, and processor for that matter like CPU or GPUs who have their own unprogrammable style of programming/architectural design which can be changed to a limited size. Hence, they adapt easily based on the given need. In this case, as part of the healthcare monitoring project, FPGAs being re-programmable targets makes them qualify more than naturally rigid CPUs/FPGA due to its modify characteristics. These new products address unique application-oriented component-level requirements to offer a fully customized solution (enabling faster time-to-market), and serve various medical applications such as patient monitoring up to real-time imaging with a very high level of functionality [25]. It is true that the natural parallel processing ability of FPGAs makes them vastly superior in terms and they can work on more than one data flow at a time, which ARM processors are not capable of. This is especially useful in healthcare applications where the ability to process large amounts of data rapidly and precisely can mean a matter of life or death. For instance, in cardiac monitoring systems, FPGAs allow the real-time interpretation of electrocardiogram (ECG) signals as they are captured - identifying critical abnormalities faster and with less error than traditional computational methods [26], [27].

In this study, throughput is reported in Giga Operations per Second (GOPS) instead of floating-point operations per second (FLOPS), as fixed-point arithmetic was adopted for efficient FPGA implementation. The use of fixed-point representation aligns with the resource-efficient and high-speed computation goals targeted in wearable healthcare devices, where minimizing complexity and maximizing throughput are crucial.

Conventional CPU and GPU based solutions often face challenges in latency, energy consumption, and real-time responsiveness due to their software-driven architectures. These limitations can critically impact time-sensitive healthcare applications such as emergency diagnostics and continuous patient monitoring. FPGA acceleration overcomes these issues by enabling hardware-customized, low-latency, and energy-efficient processing, making it an ideal choice for high-performance healthcare monitoring systems.

1.2.2 Emerging Trends in FPGA-based Solutions

An increased proliferation of more efficient monitoring systems has spurred an uptake in FPGA technology applications within healthcare. Through the integration of FPGAs alongside Artificial Intelligence and Machine Learning algorithms, one particular trend deserves mention as very high level, that is usage CNNs. This integration effectively leverages the power of healthcare devices to simultaneously carry out deep analyses like pattern recognition in medical images quite faster than before. The net gain is an improved diagnosis rate and patient outcome [28]. In wearable health monitoring, new trend in the market is wearable FPGA devices for continuous health monitoring. Gather data on patient health metrics (e.g., heart rate, blood pressure, or glucose) in real-time. These devices take advantage of the low power and high processing speed found in FPGAs to responsibly deliver information back to a medical team as soon as possible while maintaining long battery life. The shift towards portable and effective healthcare monitoring solutions highlights the importance of FPGAs in enablement for proactive and preventive medical care [29].

In addition, the era of telemedicine and remote patient monitoring has also emphasized their applications in secure data transmission with high reliability. Hardware security is crucial in protecting the privacy of patients and providing trustworthy medical diagnosis at home. FPGAs are instrumental in encrypt sensitive data during its transition from patient devices (such as smartphones) to health service providers [30]. The importance of FPGA technology in healthcare monitoring cannot be enhanced enough. As a growth engine in the world of healthcare, AI has begun to overhaul practices in medical diagnostics, patient monitoring, and treatment. With trends such as AI integration, wearable health devices, and secure telemedicine continuing to grow over the coming years, FPGA technology will inevitably be at the heart of healthcare advances. Continued R&D in this space is set to uncover new and even more innovative use-cases for FPGA technology, which are already proving their worth when it comes to healthcare monitoring applications and beyond [31].

1.3 Introduction to CNN Architectures

CNNs are at the heart of some of today's remarkable advancements in deep learning, most notably in areas where a great deal of data like images and video need to be analyzed. The usage of CNN architectures in the healthcare sector has seen a new dawn where there is an efficiency and precision level among diagnostics [32]. This section concentrates on the specific meaning of the role to be played by One-Dimensional Convolutional Neural Network (1-D CNNs) when it comes to processing healthcare data, mainly non-image shapes such as CSV files which are extensively available in the field of healthcare data analysis [33].

1.3.1 1-D CNNs and Acceleration

1-D CNNs are used to model the feature across one dimension (time series) while capturing its contextual patterns much better than other deep learning algorithms and, as a result, can be really helpful in interpreting healthcare data like heartbeats or glucose count. 1-Dimensional CNNs, on the other hand, are like their 2D and 3D versions applied in image processing but, unlike them, designed to analyze data that follows over a single dimension. It is very useful, in particular, when having to face numerous data from healthcare handled as CSV files and you need to process the information quickly [34].

The technical architecture of 1-D CNNs has been engineered to be accelerated, with the layout aligned for each layer to recognise patterns over a temporal sequence. This is accomplished by using convolutional layers that apply filters to the input data and pooling layers that reduce dimensionality, followed by a fully connected (dense) layer that interprets the features extracted earlier. The characteristic of 1-D CNNs, when tuning their hyperparameters efficiently for complex datasets, holds to a great extent (fast) in that they can easily handle very intricate data while not losing performance [35].

1.3.2 Relevance and Benefits

Utilization of 1-D CNN in the analysis of healthcare data has implications which are very significant and hence providing a solution for various problems faced by researchers currently. The use of 1-D CNNs allows the diagnostics by healthcare providers to be improved and additionally increase overall data throughput, while at the same time shortening analysis speed in situations where time-sensitive decisions can impact patient care. In addition, the use of 1-D CNNs makes it possible to discover subtle patterns in data that traditional analysis techniques cannot capture. It is a critical first step toward detecting early signs of diseases, allowing for interventions and targeted treatment plans. The sequential nature of the healthcare data is processed well by 1-D CNNs and this leads to a more informed diagnostics model which makes decision easier for physicians ensuring better patient management [36].

1.3.3 Design Considerations for Acceleration

Optimising 1-D CNN architectures for acceleration involves a series of design choices. It is important to extract relevant features out of the healthcare data, so that appropriate filter sizes and types are selected for an efficient capture with CNN. A compromise has to be made between the depth and breadth of the network for both speedup in processing as well as accuracy. When it comes to frameworks and tools, platforms like TensorFlow and PyTorch provide robust support for creating 1-D CNN models. These platforms provide a complete library and APIs using which authors can easily build 1D CNNs, providing it with a sufficient level of customisation in order to be scaled as per the specific requirements of healthcare data analysis. Moreover, the platforms provide different tools for data preprocessing, model training, and performance evaluation as well. For researchers and data scientists in the healthcare space, these are indispensable platforms [37].

This ushered a new era in actual acceleration with the 1-D CNNs for healthcare data. With their specialised architecture and processing capabilities, 1-D CNNs provide a robust framework to improve the diagnostic efficacy as well as the need for timeliness in diagnosing different diseases. The healthcare sector will no doubt keep producing volumes of healthcare data, leading to an increase in the importance and application of this information for improved patient care, treatment results, and thus expanding the utilisation/explanation ability of 1-D CNNs.

1.4 Definition and Application of Digital Twins in Healthcare

A Digital Twin (DT) in healthcare is a dynamic, real-time virtual representation of a patient, integrating data from medical records, IoT sensors, genetic profiles, and environmental conditions to continuously simulate, monitor, and predict health states. In this thesis, Digital Twin technology is employed to create cloud-integrated virtual models of patients, enabling real-time remote monitoring, predictive analytics, early anomaly detection, and personalised healthcare decision-making.

The concept of Digital Twins, originating from the fields of aerospace and advanced manufacturing, has found a revolutionary application within the healthcare sector, offering a transformative approach to patient monitoring and predictive analytics [38].

1.4.1 Definition of Digital Twins

In healthcare, Digital Twins are conceptualized as comprehensive digital replicas of individual patients or healthcare processes. These virtual models integrate real-time health data, medical history, genetic information, and environmental factors to create a multi-dimensional and dynamic representation.The health status of each patient is displayed to healthcare providers, and it can predict potential medical conditions up front for individualised treatment by providing a more precise care enforcement [39].

1.4.2 Applications in Healthcare

Healthcare use cases include patient monitoring, disease diagnosis and identification, treatment planning optimisation to improve health outcomes. For example, by using Digital Twins to simulate the progression of diseases like diabetes or heart conditions, care providers in chronic disease management are empowered with capabilities to fine-tune treatment plans and avoid pitfalls. In surgical planning, Digital Twins of organs or entire physiological systems can also be used to aid surgeons in rehearsing and planning for complex interventions, thus reducing the risks in surgery and increasing patient safety. Digital Twins further layer on top of this integration with other existing healthcare systems, pulling in data from Electronic Health Records (EHRs), wearables, and the IoT sensors designed around them. The result is an integration that diversifies data inputs into the digital twin, but one which also gives physical, genetic, and lifestyle factors equal in overall patient care [40].

1.4.3 Importance in Patient Monitoring and Predictive Analytics

There is no doubt about the importance of Digital Twins in patient monitoring and predictive analytics. Digital Twins provide a living, breathing model of the health status and condition at any point in time, making it an invaluable tool for early intervention months even years before clinical signs start to manifest. This model of healthcare leads to better patient outcomes and a reduced burden on health systems as emphasis is placed on preventive rather than episodic care few resources are misdirected due to unnecessary medications, surgeries, or clinic visits [41].

In predictive analytics, digital twins are a great way to better individualise patient care. Healthcare providers who are able to run simulated treatment scenarios and predict outcomes can now make decisions that are more tailored in accordance with the specific characteristics of each patient. With precision medicine, where treatment response can vary depending on genetic and environmental differences between patients, this level of customisation becomes all the more important [42].

1.5 Cloud Integration in Healthcare Systems

The incorporation of healthcare with cloud computing is a radical stride that enables the unification, storage, and utilization of medical data. This leads to unprecedented scale, accessibility, and efficiency in the provision of healthcare services. This section advocates the convergence strategy of cloud computing and Digital Twins, which enables predictive analytics by AI in a more comprehensive way in healthcare systems.

1.5.1 Role of Cloud Computing in Healthcare

Cloud computing materialises as a basis of innovation in healthcare, by providing an infrastructure that can scale and adapt to the extensive needs for data management within the industry. This primarily encompasses storing lots of data, then running complex processing over that quickly and making sure the results are timely. These capabilities enable healthcare professionals and patients to access critical health data ubiquitously, thereby enhancing the responsiveness and overall quality of healthcare services. Furthermore, the integration of cloud computing facilitates real-time analytics and seamless collaboration among healthcare stakeholders (e.g., doctors, hospitals, and care teams), which contributes to improved patient care coordination and clinical outcomes [43].

1.5.2 Integration with Digital Twins

The integration of cloud computing with DT technologies is transforming healthcare monitoring and predictive analytics. By harnessing the computational power and extensive storage capacity of the cloud, Digital Twins are enriched with real-time, patient-specific data, thereby increasing both their precision and predictive functionality. Cloud infrastructure enables spontaneous integration of large data sets (such as portable equipment and EHR) in DT model, facilitates more complex simulation and analysis. This extensive data provides enlarged insight into the excellent confident health professionals, thus supporting more informed and effective decision-making in the patient's care [44].

The integration of cloud computing in the healthcare system marks significant progress to take advantage of the technology to improve patient data control, analysis and care. By increasing functionality of DT and promoting future analysis features through AI, Cloud Computing creates synergistic platform that further improves innovation of the healthcare system [45].

1.6 Thesis Motivation

Healthcare faces significant challenges in delivering continuous, accurate, and efficient patient monitoring, particularly in remote environments. Traditional systems often suffer from limitations in computational efficiency, data accuracy, and energy consumption, restricting timely diagnosis and effective management of critical health conditions. Moreover, the integration of emerging technologies such as Digital Twins (DTs), Internet of Things (IoT), and Artificial Intelligence (AI) into real-time healthcare monitoring systems remains underdeveloped, particularly with respect to secure data handling and predictive analytics.

This thesis aims to address these challenges by developing a high-performance 1-D CNN accelerator for biomedical signal classification, proposing a real-time cloud-integrated Digital Twin healthcare architecture, enhancing predictive analytics through AI-driven Digital Twin frameworks, and introducing an advanced hybrid task offloading model tailored for healthcare applications. Through these contributions, the research seeks to significantly advance the efficiency, accuracy, and intelligence of remote healthcare monitoring systems.

1.7 Research Gap and Thesis Objective

Many critical gaps in the monitoring and managing of patient health remain, especially while patients are at a distance from healthcare facilities despite considerable advancements that have been achieved with the help of technology. Most healthcare systems find it difficult to integrate the advanced technologies necessary to solve complex and perennial healthcare problems in totality. In detail, the system needs to be resilient enough for monitoring patient health remotely continually with cutting-edge technologies viz cloud computing, Internet of Things, Machine Learning, and Artificial Intelligence in order to evaluate a timely as well as accurate picture of the state of health. Furthermore, existing healthcare devices often fall short in providing continuous, reliable monitoring of critical health indicators, necessitating the development and implementation of more efficient, accurate, and resource-effective devices. Challenges in managing the computational and energy demands of healthcare devices persist, highlighting the need for advanced offloading strategies that optimize network latency and energy consumption without compromising performance. Additionally, the full potential of digital twin technology in healthcare remains unrealized, particularly in integrating real-time data analytics and cybersecurity measures to enhance operational efficiency and patient care.

Addressing these gaps is crucial for the ongoing progress of healthcare technologies and the improvement of patient outcomes. Consequently, this thesis aims to confront these research challenges by introducing and implementing innovative methodologies and system architectures in healthcare technology. The core objectives are delineated as follows: Firstly, a 1-D Convolutional Neural Network (CNN) will be developed, specifically designed for the precise detection and classification of ExG signals (e.g., ECG, EMG, EEG). This model will subsequently be modified to encompass additional time-series applications, such as blood pressure and diabetes monitoring, with a particular focus on optimising accuracy and efficiency while reducing hardware resource consumption.

Secondly, a real-time architecture for remote health monitoring will be introduced, incorporating cloud computing, the IoT, ML, and AI. This architecture is intended to facilitate the continuous monitoring and evaluation of patient health through the development of an advanced DT framework, which utilises cloud-based infrastructure alongside wearable medical devices. Such integration is expected to enhance the precision of emergency notifications while enabling comprehensive real-time health tracking.

Thirdly, an advanced DT framework for healthcare will be established, addressing existing gaps by introducing a customised DT model that incorporates real-time sensor inputs, historical datasets, AI-driven analytics, and enhanced cybersecurity protocols. Novel methods for enhancing predictive analysis and identifying temporal correlations in healthcare data will be implemented and validated.

Lastly, the integration of digital twin and advanced offloading strategies in healthcare will be pursued.Partial and binary offloading strategies tailored for healthcare applications will be developed and integrated, epitomising network latency and energy consumption.The efficacy of these strategies will be demonstrated through experimental frameworks and practical implementations, supported by simulation data. By achieving these objectives, the thesis aims to significantly advance the state-of-the-art in healthcare monitoring, diagnostic precision, and efficient data processing, paving the way for future innovations in the field.

1.8 Contributions

This thesis makes four significant contributions to the field of healthcare technology through the development and implementation of innovative methods and architectures, each documented in separate research studies. These contributions are summarised as follows:

1. In Chapter 3, a 1-D CNN accelerator is presented for the detection and classification of ExG signals, including ECG beats, EMG, and EEG signals. The 1-D CNN is implemented using an efficient hardware design, which can also be adapted for other time series

applications such as blood pressure and diabetes monitoring. The introduction of the first hardware architecture designed to use three biomedical signals from ExG on an FPGA platform facilitates CNN acceleration. The architecture can compute convolution for any size of input and modify the stride value. A pipelined processing unit array is designed to achieve high performance and efficiency, including a sign bit in each processor unit to minimize power consumption and lower hardware resource costs.

In this study, fixed-point arithmetic was adopted for efficient FPGA implementation, and throughput is reported in Giga Operations per Second (GOPS) instead of floating-point operations. The proposed design achieves a throughput of 1145 GOPS at 442.948 MHz with 1.068 KLUT resource utilization, accurately identifying ExG signals using the Xilinx FPGA platform and attaining higher speeds compared to the classification of just one signal type.

2. In Chapter 4, a real-time architecture integrating cloud computing, IoT, ML, Pyomo model, and AI to remotely monitor and assess patient health is proposed. Data is transmitted to the cloud through sensors, and a virtual patient replica is used for monitoring and trend prediction from medical history. DT architecture based on cloud and healthcare wearables is proposed, demonstrated through a case study that addresses real-time monitoring challenges and enhances emergency alert accuracy. The use of ML for real-time comparison, diagnosis, and prediction is explored, ensuring consistent results by comparing seven different ML algorithms. A cost-effective DT simulation framework for twin graphs using JSON-LD and sensors for real-time monitoring and health tracking is introduced, utilizing pay-as-you-go cloud services. Furthermore, a wearable healthcare device for continuous patient monitoring is developed, monitoring indicators such as SpO2, heart rate, and body temperature. The design is validated through comparison of physical and digital data using time series insight (TSI), with latency calculations indicating relatively low values compared to previous studies.

3. In Chapter 5, An architectural design utilising real-time sensor data for vital sign monitoring (HR, SpO2, BT) is presented, combining historical data, real-time analytics, and AI for enhanced predictive analysis. The development of hybrid model, which combines XGBoost and Multilayer Perceptron (MLP), addresses challenges related to real-time and historical data. An advanced digital twin model (MDT) for healthcare is created, integrating cloud computing with AI, IoT, and robust cybersecurity to enhance operational efficiency and security is presented. By incorporating quantum security measures, i.e., QKD, and optimising resource allocation through cloud computing. Additionally, Novel Autocorrelation Analysis is introduced to enhance the identification of temporal correlations in healthcare data, detecting patterns and statistical significance in predicted disparities. An advanced rolling average method is implemented and validated on over data points, addressing variability in healthcare data and providing a reliable and precise

instrument for patient surveillance and prognosis.

4. In Chapter 6, a bespoke amalgamation of partial and binary offloading strategies tailored for healthcare applications is introduced. The integration of DT and Social Health Determinants into offloading deliberations fosters preemptive health interventions and personalised patient treatment paradigms. The pragmatic efficacy of the Digital Twin Healthcare Enhanced Asynchronous Team-Based Multi-Agent Proximal Policy Optimisation (DTH-ATB-MAPPO) within real-world healthcare settings is demonstrated, showing superiority in terms of rapid convergence and optimisation of rewards. MEC systems are refined through new aspects of DT adoption, significantly improving network latency and energy consumption. Experimental frameworks are constructed, linking theoretical groundwork with practical implementation, supported by simulation data of several operational cases. Furthermore, Adaptive Cybersecurity Task Offloading (ACTO) is innovated, using adaptive protection functions and exact matching technology to identify threats and respond with adaptive cybersecurity mechanisms without sacrificing computational and storage abilities. A new algorithm named AI-Quantum-Digital Twin-IoT (AQDT-IOT) considers the quantum pre-processing to support decision-making regarding task offloading aiming both better performance and reliability.

These contributions collectively advance the state-of-the-art in healthcare monitoring, diagnostic precision, and efficient data processing, paving the way for future innovations in the field.

1.9 Thesis Outline

In addition to the introduction chapter, this thesis includes six other chapters. Here is an overview of each chapter:

- Chapter 2 provides an overview of related work of the devices monitoring remote with explain about AI and task offloading. The chapter discusses the state-of-the-art in ExG signal detection and classification using CNNs and FPGAs, real-time remote health monitoring architectures integrating cloud computing and IoT, advanced digital twin frameworks for healthcare, and the integration of digital twin and offloading strategies in healthcare applications.

- Chapter 3 covers the development of a 1-D CNN for the detection and classification of ExG signals (ECG, EMG, and EEG). The hardware implementation using FPGA technology, the architectural design, and the performance evaluation are detailed.

- Chapter 4 focuses on a real-time architecture for remote health monitoring. This chapter explores the integration of cloud computing, IoT, machine learning, Pyomo model, and AI, describing the design of the DT architecture, the implementation of wearable healthcare devices, and the system validation through a case study.

- Chapter 5 presents an advanced DT framework for healthcare. An architectural design utilising real-time sensor data for vital sign monitoring is discussed, alongside the development of the hybrid model and the integration of cloud computing, AI, IoT, and improved predictive analysis and patient monitoring are introduced. Enhanced data integrity and privacy within the DT framework by integrating quantum security mechanisms, i.e., Quantum Key Distribution (QKD), into the DTHQ(A,B,Q) protocol, ensuring health data protection against threats posed by classical and quantum computing advancements.

- Chapter 6 explores the integration of DT and advanced offloading strategies in healthcare. The chapter introduces a bespoke amalgamation of partial and binary offloading strategies, the integration of social health determinants, and the efficacy of the DTH-ATB-MAPPO framework and quantum computing. Refinement of MEC systems, experimental frameworks, and the innovation of Adaptive Cybersecurity Task Offloading (ACTO) are also addressed.

- Chapter 7 includes a summary of the findings of this research. Conclusions drawn from the research are provided, and the implications for the field of healthcare technology are discussed. Potential directions for future research are outlined, highlighting areas where further investigation and development are needed to build on the contributions of this thesis.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, the various literatures reviewed and which are connected to the research areas of interest have been detailed. The focus areas include the design and implementation of efficient CNN architectures on FPGAs for healthcare devices, the development of cloud-based digital twin ecosystems, and innovative task offloading strategies in healthcare. The purpose of this chapter is to establish the context for the current research by examining the state of the art, identifying key trends, methodologies, and findings, and highlighting gaps and limitations in the existing body of knowledge. Through a systematic review of relevant studies and frameworks, the research will be positioned within the broader academic discourse, thereby justifying the necessity and novelty of the work undertaken. This chapter is organised as follows: The next Section 2.2, gives the outline of the review encompassing the discussion of the purposes of the present literature review and the major areas of concern in this regard. In Section 2.3, the aim of this section is to introduce and describe the background and main definitions that would be useful in carrying out the research. In Section 2.4, there is a brief summary of the main concepts and definitions that are pertinent to the analysis. According to Section 2.5, cloud-based digital twin ecosystems. In Section 2.6 identifies more details in the way IoT and digital twins can be integrated. In Section 2.7, the task offloading strategies in healthcare are discussed. Comparison of the existing work in Section 2.8. Finally, Section 2.9 is located in the context of the reviewed literature, and Section 2.10 draws the chapter to summarise the key points.

2.2 Scope of the Review

The scope of this review encompasses a comprehensive examination of the existing literature pertinent to several key areas within the intersection of advanced computational methods and healthcare technology. The primary focus areas of this review include:

1. CNN Architectures on FPGAs for Healthcare Devices:
    - Exploration of the design and implementation of CNN architectures on FPGAs.
    - Assessment of the efficacy of CNNs in medical applications, particularly in signal processing and real-time data analysis.
    - Evaluation of key studies that highlight advancements, methodologies, findings, and impacts on healthcare diagnostics and monitoring systems.
    - Identification of limitations and gaps in existing FPGA-based CNN implementations.
    - Research Question: How can CNN architectures on FPGAs be optimised to improve the accuracy and efficiency of medical diagnostics and real-time monitoring? The review will explore existing CNN implementations on FPGAs, focusing on their performance in processing biomedical signals such as ECG, EEG, and EMG. The potential of these architectures to enhance diagnostic accuracy and processing speed will be assessed, along with challenges in scalability, resource utilisation, and integration with existing healthcare systems.
2. Cloud-Based Digital Twin Ecosystems:
    - Definition and conceptualisation of digital twins within the healthcare context.
    - Analysis of applications of digital twins for real-time monitoring, predictive analytics, and personalised treatment plans.
    - Review of significant studies and contributions in the development and implementation of cloud-based digital twin ecosystems.
    - Discussion of challenges, such as data standardisation, real-time synchronisation, integration with IoT devices, and identification of research gaps.
    - Research Question: What are the most effective strategies for integrating digital twins with IoT devices to enhance real-time monitoring and predictive analytics in healthcare? This question seeks to understand the current methodologies used to link digital twins with IoT technologies, examining their applications in chronic disease management, real-time health monitoring, and personalised healthcare.
3. Task Offloading Strategies in Healthcare:
    - Emphasis on the importance of task offloading for enhancing computational efficiency, reducing latency, and conserving energy in healthcare applications.
    - Review of existing algorithms and strategies, including heuristic and machine learning approaches, and their effectiveness in healthcare contexts.
    - Comparative analysis highlighting the advantages, limitations, and areas requiring further research in task offloading strategies.

- Research Question: How can task offloading strategies be optimised to balance computational efficiency, energy conservation, and security in healthcare applications? This review will analyse various task offloading strategies, focusing on their ability to manage workloads efficiently while ensuring data security and minimising energy consumption. By comparing heuristic and machine learning-based approaches, the review aims to highlight best practices and identify gaps that need addressing to improve the overall effectiveness of task offloading in healthcare.

4. Integration of Internet of Things and Digital Twins:
   - Investigation of the role of IoT technology in healthcare and its synergistic integration with digital twin technology.
   - Evaluation of key studies demonstrating the potential of IoT-DT integration for enhanced monitoring, simulation, and prediction of health conditions.
   - Analysis of current challenges and gaps, such as data integration complexities, privacy and security concerns, real-time data integration, and cost and resource constraints.
   - Research Question: What are the key challenges and potential solutions for integrating IoT technology with digital twin systems to improve healthcare outcomes? The review will explore the synergies between IoT and digital twin technologies, assessing their combined potential to transform healthcare monitoring and predictive analytics. Key challenges related to data integration, privacy, security, and cost will be examined, with the aim of identifying innovative solutions that can enhance the implementation and effectiveness of IoT-DT systems in healthcare.

By encompassing these focus areas, the review aims to establish a comprehensive understanding of the current state of the art, identify critical trends and methodologies, and highlight gaps and limitations within the existing body of knowledge. This foundational review sets the stage for the proposed research by justifying its significance, novelty, and potential contributions to the field of healthcare technology.

2.3 Background and Overview

This section provides a concise background and introduction to the key technological domains relevant to this research, including Convolutional Neural Networks (CNNs), Field-Programmable Gate Arrays (FPGAs), DT, task offloading strategies, and IoT integration within healthcare contexts. These foundational technologies form the basis for developing efficient, real-time, and adaptive healthcare systems.

CNNs emerged in the 1980s with early models such as the Neocognitron [46], and experienced major resurgence after the success of AlexNet in 2012 [47], demonstrating the potential of deep learning for complex pattern recognition. In healthcare, CNNs have been successfully applied to medical imaging, biomedical signal processing, and disease diagnosis.

Digital twin technology, initially proposed by Grieves in 2002 for manufacturing applications [48], has since evolved into healthcare, providing virtual representations of patients or medical devices to support personalized monitoring, diagnosis, and predictive analysis.

Task offloading, originating from distributed computing paradigms, has become critical in modern networked environments. In healthcare, task offloading enables lightweight medical devices to delegate computationally intensive operations to more capable edge or cloud infrastructures, enhancing system responsiveness and conserving local resources.

2.4 Review of Key Areas

The advancement in healthcare technology has dramatically changed through the use of complex computation models as well as effective hardware solutions. This section briefly highlights important areas related to the study and with emphasis on CNNs and FPGAs. CNNs, their use in medical domain and especially signal processing and real-time data processing is examined together with the FPGA advantages.

Firstly, a general introduction to CNNs is given, including the description of the network's architecture and purpose in biomedical signal processing. The following section provides a deeper analysis of the application of CNNs on FPGAs in the targeted area of healthcare to analyze how such integration results in increased effectiveness and precision of the medical diagnosis and monitoring systems.

Key studies in the field are analysed, focusing on their methodologies, findings, and impact on healthcare technology. This includes a detailed examination of CNN applications in processing EEG, EMG, and ECG signals. Each subsection addresses specific advancements, challenges, and outcomes of these studies.

Finally, limitations and gaps in existing research are identified, highlighting areas where further investigation is needed. By reviewing these key areas, foundation is set for the proposed research, justifying its significance and potential contributions to the field of healthcare technology.

2.4.1 CNN Architecture on FPGA

A. Overview of Convolutional Neural Networks

Many real-world patterns represent nonlinear alterations of their initial unique shapes. For example, in optical image categorization, objects such as vehicles and trees exhibit distinct sizes, colors, textures, locations, and perspectives. Backgrounds, non-relevant items, and varying light intensities further complicate the challenge. These alterations can render objects indistinguishable in vector space. Convolutional and pooling layers in CNNs filter and scale effective features to address this problem. The concept space is split or transformed using one or two dense layers, and the needed labels or the target map is revealed in a classification or regression task [49], [50].

In CNNs, the convolution layers reproduce a version of the convolution operation commonly found in Linear Time-Invariant (LTI) systems. In the 1-D case, which is particularly suitable for sequential data such as biomedical signals, the convolution of an input signal x with a filter w and a bias b is expressed as:

$$(x \times w)(i) = \sum_{m=0}^{k-1} x(i+m)\, w(m) + b \tag{2.1}$$

where k is the size of the filter. This operation extracts local patterns from the input sequence by sliding the filter along one dimension and aggregating weighted sums. Properly learned filters amplify significant signal features while suppressing noise or irrelevant information [51]. Although originally developed for two-dimensional inputs, CNN architectures have been successfully adapted for one-dimensional and three-dimensional data structures, including time-series biomedical signals and volumetric medical imaging [52].

1. Mathematical Foundations of CNNs

Convolutional Neural Networks extract hierarchical features by applying a series of learnable transformations, primarily through convolutional, activation, pooling, and fully connected layers. In this section, the mathematical operations are presented specifically for one-dimensional CNNs (1D-CNNs), which are particularly suited for sequential data such as biomedical signals.

a) Convolutional Layer: Given an input signal $\mathbf{x} \in \mathbb{R}^{L_{in}}$ and a convolutional filter $\mathbf{w}_m \in \mathbb{R}^k$ for the m-th output feature, the 1D convolution operation produces an output sequence $y_m[i]$

as:

$$y_m[i] = \sum_{h=0}^{k-1} x[i+h]\, w_m[h] + b_m, \tag{2.2}$$

where b_m is the bias term, k is the kernel size, and i denotes the index along the input. Stride s defines the step size of the filter movement, and padding p controls zero-padding around the input. The output length L_{out} is computed as:

$$L_{out} = \left\lfloor \frac{L_{in} + 2p - k}{s} \right\rfloor + 1 \tag{2.3}$$

Here, L_{in} is the length of the input sequence. Padding p increases the effective input size, and the floor operation ensures that only full filter applications are counted, with an additional 1 accounting for the initial position.

b) Activation Function: Activation functions introduce non-linearity into the network, enabling CNNs to model complex and non-linear patterns. In 1D CNNs, activation functions are applied independently to each element of the output feature sequence. A common choice is the Rectified Linear Unit (ReLU), defined as $\sigma(x) = \max(0, x)$, which sets all negative values to zero and retains positive values, promoting sparse activations. Another activation is the Sigmoid function, expressed as $\sigma(x) = \frac{1}{1+e^{-x}}$, which squashes input values into the range $(0, 1)$ and is particularly useful for probabilistic outputs. The Hyperbolic Tangent (Tanh) activation, given by $\sigma(x) = \tanh(x)$, maps inputs to the range $(-1, 1)$, providing zero-centered activations that often improve convergence. The resulting activated output is denoted as:

$$a_m[i] = \sigma(y_m[i]), \tag{2.4}$$

where $a_m[i]$ is the activation at position i in the m-th feature map.

c) Pooling Layer: Pooling layers in 1D CNNs are employed to reduce the temporal dimension of feature maps, thereby decreasing the number of parameters and computational load, while preserving important features.

In 1D max pooling, the maximum value within a local window of size f is selected, as shown in Equation 2.5:

$$p_m[i] = \max_{h=0}^{f-1} a_m[i \cdot s_p + h] \tag{2.5}$$

Alternatively, 1D average pooling computes the mean value within each window, given by Equation 2.6:

$$p_m[i] = \frac{1}{f}\sum_{h=0}^{f-1} a_m[i \cdot s_p + h] \tag{2.6}$$

The output length after pooling, L'_{out}, is calculated as:

$$L'_{out} = \left\lfloor \frac{L_{out} - f}{s_p} \right\rfloor + 1 \tag{2.7}$$

where L_{out} is the length of the feature map after convolution and activation, f is the pooling window size, and s_p is the stride of the pooling operation.

d) Fully Connected Layer: After the convolutional and pooling operations in a 1D CNN, the extracted high-level temporal features are flattened into a one-dimensional vector $\mathbf{z} \in \mathbb{R}^{L'' \cdot C''}$, where L'' is the final temporal length and C'' is the number of output channels.

This flattened vector is fed into a fully connected (dense) layer, where each neuron computes a weighted sum of all input features followed by a bias addition, as described in Equation 2.8:

$$\mathbf{o} = W\mathbf{z} + \mathbf{b}, \tag{2.8}$$

where $W \in \mathbb{R}^{N_{out} \times N_{in}}$ is the weight matrix, $\mathbf{b} \in \mathbb{R}^{N_{out}}$ is the bias vector, and $\mathbf{o}$ is the output vector of the fully connected layer.

For classification tasks, a softmax function is typically applied to the output $\mathbf{o}$ to convert it into a probability distribution over the classes, as shown in Equation 2.9:

$$\text{Softmax}(o_i) = \frac{e^{o_i}}{\sum_{j=1}^{N_{classes}} e^{o_j}} \tag{2.9}$$

where o_i denotes the score corresponding to the i-th class.

e) Training via Backpropagation: The training process of a 1D CNN involves optimizing the learnable parameters θ (such as convolutional weights, biases, and fully connected weights) by minimizing a loss function that measures the difference between the predicted output $\hat{\mathbf{y}}$ and the ground truth labels $\mathbf{y}$.

For classification tasks, the cross-entropy loss is commonly employed, as shown in Equation 2.10:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{N_{classes}} y_i \log(\hat{y}_i) \tag{2.10}$$

Alternatively, for regression-based tasks or signal reconstruction, the mean squared error (MSE) is used, as described in Equation 2.11:

$$L(y, \hat{y}) = (y - \hat{y})^2 \tag{2.11}$$

The network parameters are updated iteratively through the gradient descent optimization algorithm, where each parameter is adjusted proportionally to the negative gradient of the loss function. The general update rule is given in Equation 2.12:

$$\theta_{t+1} = \theta_t - \alpha \nabla_\theta L \tag{2.12}$$

where α is the learning rate, and $\nabla_\theta L$ represents the gradient of the loss with respect to θ.

2. 1D CNNs for Biomedical Signals

CNNs configured in one dimension are effective for processing sequential data such as ECG, EEG, and EMG signals. These models apply temporal convolutions and pooling operations, allowing real-time detection of temporal patterns while reducing feature dimensionality. This architecture enhances computational efficiency and diagnostic performance in digital health applications.

3. FPGA-Based CNN Acceleration

Figures 2.1 and 2.2 illustrate CNN accelerators implemented on FPGA. The matrix multiplication engine (MME) performs all CNN tasks, including convolution and pooling. Biases are stored in registers, and input/weights are loaded via ping-pong buffering to minimize latency. Quantization to 16-bit fixed-point formats improves hardware efficiency while maintaining accuracy [53]. FPGA accelerators offer reconfigurable, low-power processing, suitable for real-time healthcare systems [54].

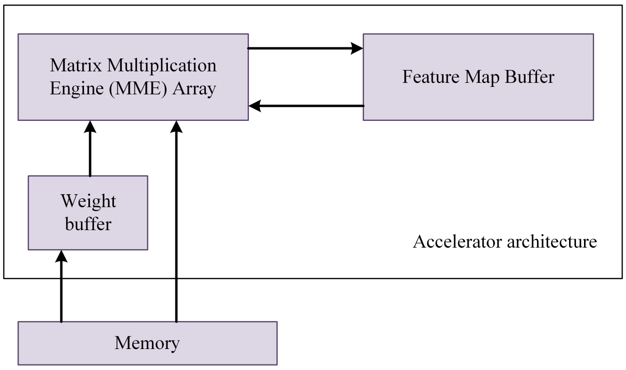


Figure 2.1. Block diagram of accelerator [55].

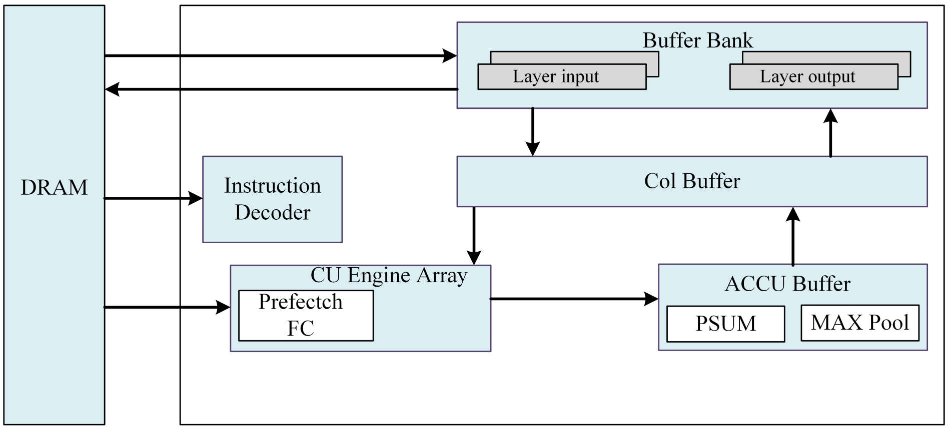


Figure 2.2. Overall architecture of the CNN accelerator [53].

B. Key studies and their findings

1. ECG Signal

The ECG devices are employed in clinical practice to measure electrical activity of the heart and help in the diagnosis of different forms of heart ailments. Electrodes are then positioned on the patient's body in order to gather the required information [56]. The advancement of wearable sensors has enhanced the capability to monitor and analyse different aspects of ECG signals [57]. Additionally, real-time monitoring systems employ diverse methods to identify ECG patterns [58]. The CNN accelerator is composed of three primary components: memory, PE array, and control logic. Figure 2.3 demonstrates the interaction among these components. Initially, the control logic fetches data from off-chip memory into the on-chip memory. Subsequently, the data is moved from the on-chip memory to the PE array for processing. The results of these computations are temporarily stored in the on-chip memory before being transferred back to off-chip memory [59], [60]. In recent developments, the IoT has enabled remote patient monitoring. Through learning from cardiology datasets, AI algorithms are applied to classify and identify diseases with high speed and high accuracy. Specialized processors aid in this process [61], [62]. Jiahao et al. presented an efficient hardware architecture for ECG classification using 1-D CNN with global average pooling for classification. This design, worked on the FPGA Xilinx

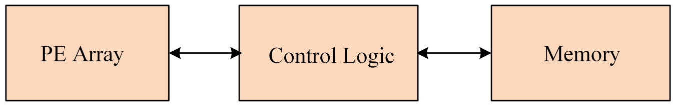


Figure 2.3. Relationship between modules of the CNN accelerator [59].

Zynq, had an average processing rate of 25.7 GOP/s at 200 MHz, implemented with 1538 LUTs and achieved substantial resource utilization enhancement. The pipelined processing unit array enhanced calculation speed, achieving an ECG beat classification accuracy of 99.10% [63]. Guo et al. described the Angel-Eye, a flexible CNN accelerator architecture that incorporates a data quantization technique and a compilation tool. This approach reduces bit width from 16 to 8 bits while maintaining accuracy, and the compilation tool adapts the CNN model to the hardware efficiently. Tests on the Zynq XC7Z045 platform showed that Angel-Eye operated at a higher speed of 150 MHz, utilizing 183 KLUTs with a power consumption of 9.63 W, and offered up to 16% improved energy efficiency compared to equivalent FPGA models [64]. FPGA based accelerator architecture was designed by Gong et al. which was implemented with synchronous pipeline instruction system. This model consisted of more specific enhancements aimed at increasing computational efficiency, while performance achieved on platforms such as Xilinx Zynq-7020 and Virtex FPGA, with enhanced speed and at the same time 2.15 W power consumption and utilising 38.136 KLUTs [65].

2. EEG Signal

EEG can be used as an assessment of emotions in order to improve the social integration of human patients with early stage Alzheimer's disease and neurological disorders. Traditionally, emotions have been classified using software on computers without internet connectivity. However, the use of wearable classifiers is essential to improve patients' social lives. In ref [66], proposed the BioCNN which is a hardware CNN for biomedical application with especial focus on EEG-based emotion detection. The training technique was part implemented using Digilent Atlys Board and Spartan-6 FPGA which is comparatively affordable than other instruments. The obtained performance results were 100 MHz, and the reduced resource usage of KLUT and improved resource usage efficiency followed. The architecture of a CNN accelerator for EEG signals is illustrated in Figure 2.4. Key components are a PE array with local scratch pads and MAC units, a Global Buffer (GLB) interfacing with off-chip DRAM, and a Network-on-Chip (NoC) for efficient data transfer. Compression and ReLU modules enhance energy efficiency for real-time neural data classification [67].

In EEG categorisation the two major tasks are to identify epileptic seizures and to distinguish between different emotional states. Creating a dedicated energy-efficient on-body seizer detector may help alert nearby people about the situation or apply an appropriate stimulus to stop the seizer. Precision is crucial for accurate seizure detection to ensure safety [68]. In ref [69], demonstrated that scalp multichannel signalling and electroencephalography are effective for real-time epileptic seizure detection. They created a novel architecture that extracts additional features with high accuracy and speed, implemented on the FPGA platform Virtex-5. Their architecture showed consistent and reliable detection and identification of epileptic episodes, achieving low resource usage in KLUT.

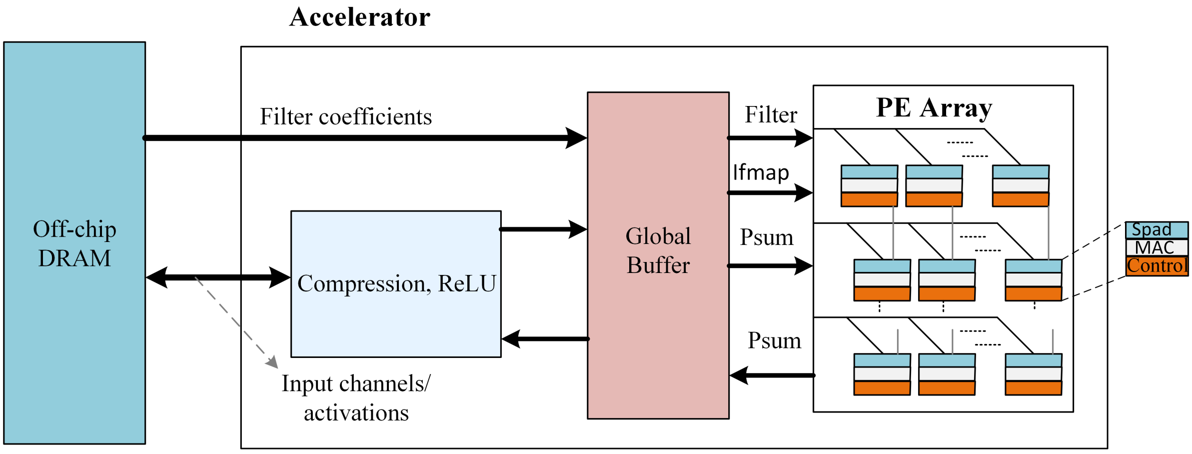


Figure 2.4. Architecture for CNN accelerator for EEG proposed in [67].

3. EMG Signal

In recent time, because of multiple uses in gesture recognition applications, more attention has been paid towards EMG processors. Due to the trade-off between the classification efficiency and power consumption, they integrate wearable devices for gesture recognition. In [70], the author developed a low power embedded system for EMG acquisition and gesture recognition mainly working on the multilevel designs of the software and hardware. Other than EMG sensors, inertial and pressure sensors were also employed for improving the gesture recognition and also the motion tracking [71].

The availability of specialised CNN accelerators has led to new possibilities of edge healthcare and biomedical processing [72]. In ref [73], developed a method of analysing surface Electromyography (sEMG) to know how the nervous system controls muscle contraction. There was developed the FPGA-based real-time Negative Matrix Factorization (NMF) processor, which extracts muscle synergies from the 8-electrode EMG signals to be classified by the SVM. The realization of them on an FPGA platform indicated the improve speed, low resource consumption in KLUT, and low power dissipation. Described and implemented an architecture in ref [74] to estimate the desired clench strength of the hand using EMG signals, which was implemented on the Xilinx XC7Z020. Their architecture showed evidence of the practical use it has in applications regarding prosthetics. The architecture realized a speed of high in MHz, the usage of resource of KLUT was low, and the power was low in W.

These FPGA implementations cover different types of machine learning models 1-D CNN,

Table 2.1. Overview of FPGA Implementations for Various Machine Learning Models.

Ref.	[63]	[1]	[75]	[74]	[64]
Method	1-D CNN	1-D CNN	2-D CNN	ANN	VGG-16
FPGA	Zy xc7z045	Zy xc7z045	UltraPlus iCE40UP5k	Zy xc7z020	Zy xc7z045
Data Size (bit)	Fixed 16, 32 Fp	Fixed 16, 32 Fp	Fixed 16/8/1	Fixed 16	Fixed 16, 32 Fp
Frequency (MHz)	200	442.948	0.1	388.20	150
DSP	80	9	8	25	780
KLUT	1.538	1.067	2.8	4.379	183
Throughput	16 GOPS	1145 GOPS	0.8 GOPS	9.705 GOPS	117 GOPS

2-D CNN, ANN, and VGG-16—as presented in Table 2.1. Each entry lists the type of FPGA, data size, operating frequency, number of DSPs, and performance in throughput. Focusing on the 1-D CNN implementations on the Zy xc7z045 FPGA, two cases are highlighted. In the first case [63], an operating frequency of 200 MHz with 80 DSPs and 1.538 KLUT achieved a throughput of 16 GOPS. In the second case [1], the operating frequency is significantly higher at 442.948 MHz, with only 9 DSPs and 1.067 KLUT, achieving a much higher throughput of 1145 GOPS. For the 2-D CNN implementation [75], the UltraPlus iCE40UP5k FPGA was used, operating at a very low speed of 0.1 MHz with 8 DSPs and 2.8 KLUT, resulting in a throughput of 0.8 GOPS. In the ANN implementation [74], the Zy xc7z020 FPGA was used, achieving 388.20 MHz operating frequency with 25 DSPs and 4.379 KLUT, leading to a performance of 9.705 GOPs. Lastly, the VGG-16 implementation [64] employed the Zy xc7z045 FPGA, utilizing 780 DSPs and 183 KLUT, achieving 117 GOPS throughput. This comparative overview highlights how the resource allocation (DSPs, LUTs) and design choices significantly affect throughput performance across different ML models on FPGA platforms.

2.4.2 Limitations and Gaps in Existing Research

Despite the increasing deployment of CNNs on FPGAs for healthcare signal processing, several domain-specific limitations persist that hinder clinical translation:

- **Lack of support for multichannel biomedical signals:** Many FPGA-based CNN designs are tailored for single-channel data, limiting their applicability to real-world multi-lead ECG or multi-channel EEG/EMG systems.
- **Limited adaptability to patient-specific variation:** Existing FPGA accelerators lack dynamic reconfigurability to accommodate differences in patient physiology, which is critical for personalised diagnostics.
- **Latency vs. accuracy trade-off:** Achieving real-time inference with high diagnostic accuracy remains a challenge, particularly in energy-constrained environments like wearable or remote devices.

- **Inadequate clinical dataset benchmarking:** A number of studies rely on proprietary or synthetic datasets, reducing generalisability to actual clinical settings and limiting comparative analysis.

- **Poor integration with clinical workflows:** Most prior implementations are evaluated in isolated hardware setups, without end-to-end validation within live monitoring systems or hospital-grade platforms.

While previous studies have demonstrated the feasibility of CNN acceleration on FPGAs particularly for ECG, EEG, and EMG signals, these limitations restrict their scalability, clinical applicability, and trustworthiness. To address these gaps, the current research proposes a lightweight, patient-adaptive 1D-CNN architecture optimised for FPGA deployment. The model balances processing speed, energy efficiency, and classification accuracy, and is evaluated using publicly available clinical datasets. Additionally, the design is tailored for integration within real-time digital health pipelines, thereby contributing toward practical and scalable biomedical FPGA solutions.

2.5 Cloud-Based Digital Twin Ecosystems

DT is the exact digital counterpart of a physical entity characterized by dynamic and mutual interactivity. This model makes it possible for data exchange to happen concurrently between the virtual and the actual physical counterpart whereby whatever that is done physically is reflected in the virtual environment and the vice versa as shown in Figure 2.5. DT is defined as a differentiation based on the integration of five sophisticated technologies. In other words, it implies gathering both structured and unstructured information, sending this information, turning it into intelligent information, enabling users to engage with the data, and protecting it [76], [77].

literature review of 2018, mentioned in [78], the authors point out the need to discover the fundamental technologies constituting DT. Digital twins facilitate the supervision, analysis, and improvement of physical entity functions and provide updates about the performance quality to support the goals of raising well-being [79], to realize the above advantages certain measures have to be taken. For constant surveillance and establishing an online duplicate of the framework, it is crucial to have IoT, as presented in Figure 2.6. New developments in IoT mean that information may be gathered continuously through sensors that are located on the actual subject. This continuous tracking also incorporates information from social media, healthcare records, business information technologies, and other sources [80], [81].

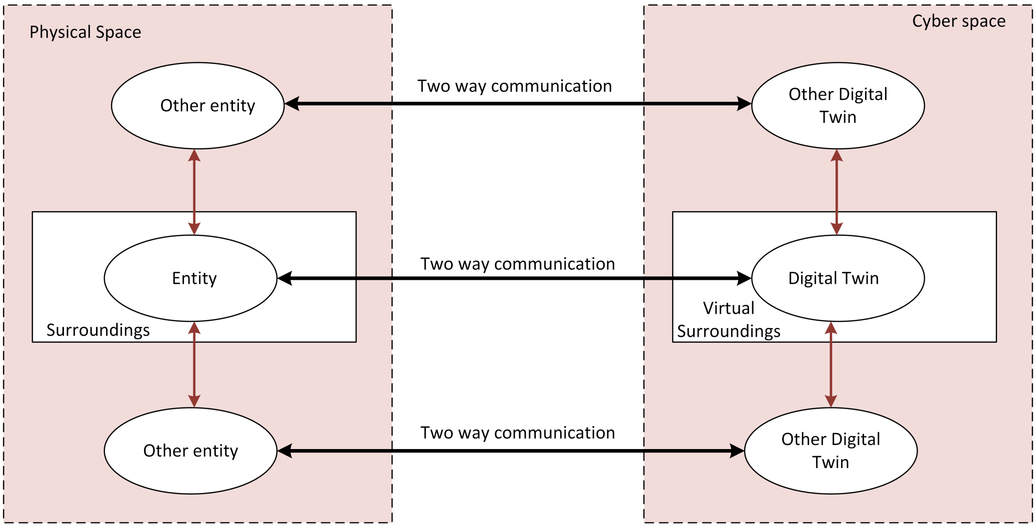


Figure 2.5. Augmented digital twin conceptual model [76].

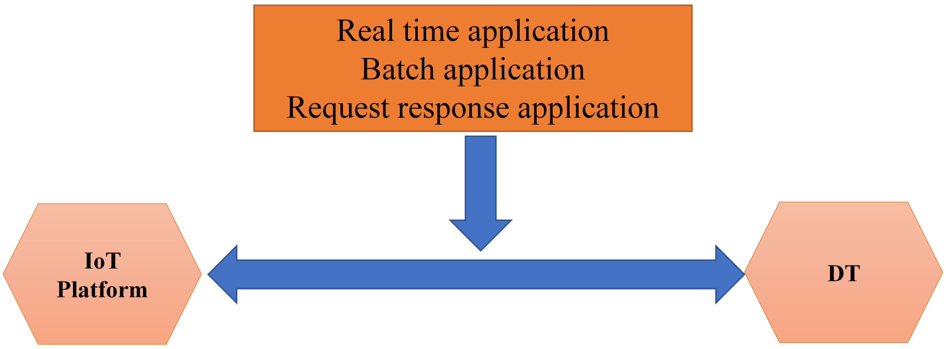


Figure 2.6. Data flow and IoT interaction with DT.

Figure 2.7 illustrated is a basic view of DT, which unveils its general characteristics and functionalities. Data is captured from physical object where a copy is developed inside a DT ecosystem reflecting the object's current state. This process is not just the documentation of raw data; the latter is paired with data analysis to reveal the object's state [82], [83].

In healthcare, DTs represent a revolutionary approach to various issues including early diagnosis of diseases and constant follow-up of diseases throughout the patient's life. For example, observing the DT of a patient with a heart disease issue lets the healthcare personnel know the chances of having a related episode, so that appropriate measures can be taken on time. It has been projected that the future use of DTs shall considerably influence the personalised therapies and intervention [84], [85]. Technology and services have found a utility in digitizing healthcare provision and functions for the healthcare professionals, and the patients as they enhance data gathering, orderly clinical communication, disease management and other related tasks [86]–[88]. In the same respect, DTs have the ability overcome limitations of existing models of healthcare, for example, in delivering patients' records in emergency situations, or in a situation where communication lines and technology infrastructure is compromised in remote

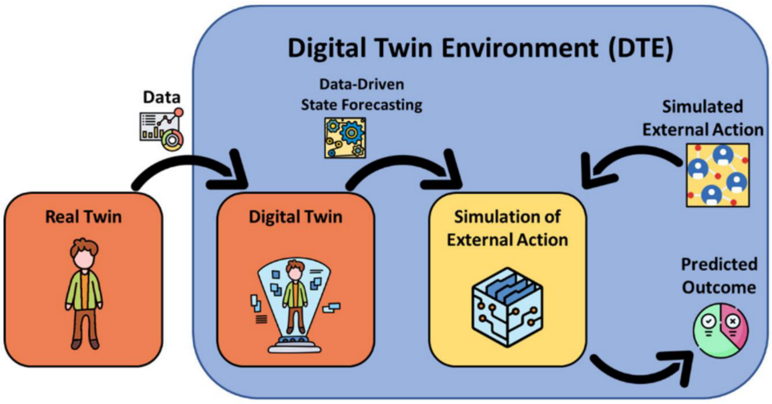


Figure 2.7. Overview of the Digital Twin platform [83].

areas. This has the benefits of leveraging on cross diagnostics and real-time medical actions [89].

DTH's main objective is to create a digital copy or a replica of human attributes such as the organs through create with a digital method of life. However, despite the supplementing of cardiology techniques, the use of DTH is still quite limited and the overall application for the promotion of the general health of the population is expected to require a few years [90]. However, the monitoring, optimizing and planning functions of DT's enable them to significantly contribute to the improvement of population health, within public health governance frameworks [91].

2.5.1 Review of Existing Cloud-Based Digital Twin Frameworks

The process of digitalization that has recently affected the majority of fields, such as healthcare, was started with the advent of the so-called industry 4.0 project in 2013. This transformation is based on the sophisticated technologies like IoT, cloud and edge computing, AI, and big data analytics [92]. The DT paradigm, which stems from these technologies, enables the digitization of any system and is applied in numerous industries, including industrial and engineering ones. DT technology in the last decade has been recommended for healthcare use and between all the implementations of this technology DTH as been one of the most remarkable works [93].

Research in DT also includes enhancing WBANs and deploying advanced signal processing and sensors that support DT, along with integrating Markov decision processes (MDP) and AI to improve efficiency and reliability in health monitoring systems [18]. Within smart homes, DT applications aim to improve the monitoring, prediction, and control of health parameters using various wireless and wearable technologies [94], [95]. In healthcare, professionals utilise DT in conjunction with cloud and IoT-edge computing, blockchain, and ML to deliver intelligent predictive diagnostics and secure health data management, thereby significantly advancing monitoring capabilities [45], [96].

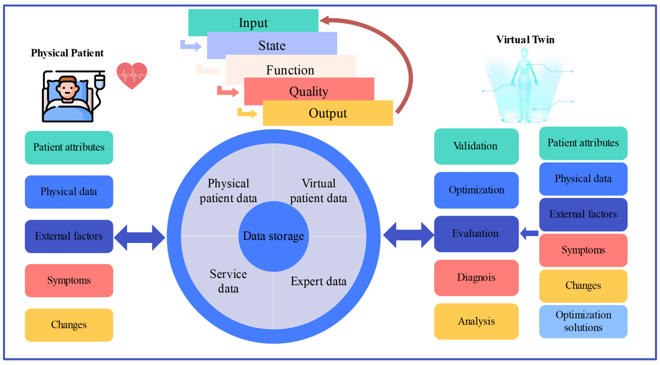


Figure 2.8. Architecture of using DT for healthcare monitoring [45].

Moreover, DT has been employed to develop frameworks that aid in clinical monitoring, evaluate patient needs, and promptly identify emergency risks [97]. For example, the Lung-DT framework integrates AI with historical radiological data and IoT sensor inputs to accurately classify lung diseases, enhancing traditional diagnostic methods. However, challenges such as the absence of data storage within this framework pose hurdles for end-users [23]. 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 [24], [98]. In Table 2.2 depicted

Table 2.2. Overview of Case Studies Utilising Advanced Technologies of DT

Ref. No.	Utilised Technology	RTD	Case Study
[99] 2023	DT, IoT, DL	Yes	Disease detection and smart medical service
[100] 2023	DT, Decentralised learning with blockchain.	Yes	Industrial ecology learns through data and resource sharing.
[101] 2021	DT, IoT, DL, ML	No	DT and IoT could revolutionise Healthcare
[102] 2022	DT, Cloud computing	N/A	VR Cloud DT Framework

application of DT that in Ref. [99], technologies e.g., DT, IoT, and DL were utilized to develop a case study focused on disease detection and smart medical services. Similarly, Ref. [100], DT and decentralised learning with blockchain were employed to examine how industrial ecology can learn through data and resource sharing. Also, Ref. [101] considered the use of DT, IoT, DL, and ML, although real-time data (RTD) was not utilised, to explore the potential revolution in healthcare. Lastly, in Ref.[102], DT and cloud computing were applied to create a virtual reality cloud framework for interactive DT applications. Through these studies, the versatility and potential of various advanced technologies in different fields were demonstrated, highlighting their significant contributions to innovation and efficiency.

2.5.2 Studies on the Integration of IoT Devices with Digital Twins

Grieves introduced a comprehensive framework for the DT model, which is structured in three dimensions: materialized, digitalized, and link [103]. Tao et al., extended this to a five dimensional architecture with integration of DT related information and resource [104]. To this end, DT applications in healthcare include big data integration and the use of AI models to mimic human physiological systems and suggest apropos clinical approaches; however, it presents technology, privacy, and ethical concerns [105]. Yang et al., attempted a new DT paradigm in which cardiovascular casts, CT scans and simulation algorithms were used to control ultrasonic probes in virtual modes [106]. This framework focuses on creating high-fidelity Cardiac Electrophysiology DT by using clinical 12-Lead ECGs. But it has no combination with cloud computing and IoT for monitoring and concentrates exclusively on the aspect of cost [107].

In Ref. [108], the automated gait data control system for fully actuated lower limb exoskeleton DT in the medical rehabilitation application is introduced. In previously published work Ref. [99], the authors analyzed the use of deep learning and big data analytics in digital twin of healthcare for real-time health monitoring. Hence, though progress was made in making devices internet-ready, cost and latency were not sufficiently solved. Liu et al., therefore, designed a cloud-based framework for elderly healthcare using DT technology known as CloudDTH that requires the implementation of big data, cloud computing, and IoT. But, data latency in real-time data and its cost were not expounded adequately, as well as the use of the ML component in predictive capability for the HC DTs [109].

ESP32 Azure IoT Kit which is supposed to be compatible with Microsoft's cloud platform consists of an extensive list of on board sensors as shown in Figure 2.9, suitable for healthcare-oriented digital twins' development. These sensors can provide an opportunity to receive the most important measures of physiology, environment, and motion that are critical to create an objective model of a patient or healthcare environment [82].

Incorporating a network of sensors within a typical healthcare monitoring system (HMS) ecosystem helps in remote sensing and instantaneous measurement of paramount parameters relevant to the contemporary status of any structure and its surroundings. Some of these parameters include Strains, stress, Humidity, temperature and other important ones that would enable constant health checks to be carried out on the structure.

In digital twin healthcare, there are several sensors are implemented:

- Motion Sensor (MPU6050): In the case of a digital twin of healthcare, motion data can be used to track the changes in the patient's status in terms of the ability or the inability

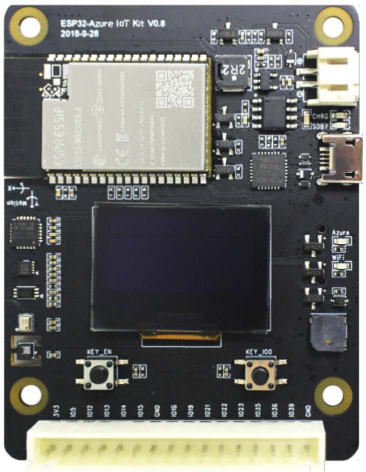


Figure 2.9. **The ESP32 Azure IoT kit.**

to move, the occurrence of falls, and the distribution of activity throughout the day.

- Magnetometer (MAG3110): This sensor is used to measure magnetic fields and thus which do the work of a compass. Even if it is most often used to determine orientation, which could offer possibilities for navigation in healthcare facilities if magnetically tagged medical equipment is transported or if exposure to electromagnetic fields is measured.

- Barometer (FBM320): In context of DT healthcare, collecting this data into a digital twin could enable identifying individual-specific relations between variations in pressure and patients' complaints.

- Humidity & Temperature Sensor (HTS221): This combined sensor can accurately detect the humidity level and temperature two variables that influence the patients' comfort and health condition. Temperature and humidity data in a home environment can be beneficiary to the patients with respiratory disorders or allergies.

- Light Sensor (BH1750FVI): In digital twin healthcare, used sensor allows for evaluation of light and its impact on patient outcome (e.g., mood and sleep) and enhance well being.

The inclusion of various sensors such as barometers, magnetometers, and motion detectors in this review is intended to illustrate the full range of environmental and physiological data sources available for future digital twin healthcare applications. While not all sensors are directly used in the current implementation, understanding their roles and potential integration pathways helps identify how diverse sensor data can enhance patient modeling, environmental context awareness, and real-time predictive analytics in future system extensions.

2.5.3 Relevance to Digital Twin Healthcare

The ESP32 Azure IoT kit provides a diverse capability of the sensors, which forms the foundation of obtaining the three-dimensional data that will be crucial in developing accurate and comprehensive digital twins in the healthcare fields. From the above sensors, various medical practitioners can be able to derive more comprehensive information on patient physiology, environment and movement, which can help them develop models, treatments, utilizations and even improvements for the existing health system [82], [110]. Additionally, the MAX30102 and MLX90614 external sensors were connected to the input pin of ESP32-Wrover-B, and real-time data could be monitored if connected to the worldwide web [111], [112].

2.5.4 AI Models in DT for healthcare

It is dynamically enabled for monitoring, understanding, as well as improving the human health indices in terms of DT technology, thereby providing constant guidance to advance the quality of life [113]. Studies in the area of DT have since in the year 2018-2024 garnered significant research interest [114]. DT models were first used as an approach to the acquisition, processing, and visualization of data [115], [116], which include, e.g., the real-time status of one's heart condition. The focus moved to well-being systems with the help of AI since DT can use AI for data processing and such activities as decision-making, predicting, and others. For example, a DT health monitoring system can monitor a patient's heart rate on an emergency basis with live feedback [116], whereas AI can notify of heart rate abnormalities [117], [118].

The application of AI in the healthcare is not a new idea. It is already possible about the improvement of the human quality of living through the use of AI. In IoT and Cyber-Physical Systems (CPS), AI is used in decision making and predictive functions [119]. First of all, during the DT revolution in the heathcare sector the major goal was, for instance, the provision of continual health monitoring. Another DT healthcare model was described in [120] for the elderly health monitoring where the data was collected and transmitted through IoT technology and analyzed in the cloud. While with the help of this model, it is possible to perform virtual health monitoring of seniors in real-time, the application of automated decision-making was not provided. Specifically, human input was needed in order to discern alerting information concerning a patient's health status that could include a heart rate, pulse rate, etc; or alerts of a looming crisis. The said vogue of operation in the model was to possess a twofold effect: the inability to foresee extreme occurrences limited the model to real-time data analysis only. Another study [114] aimed at the problems regarding the heterogeneity of collected data and suggested a solution in the form of ISO/IEEE 11073 DT for health and well-being. This framework involves:

- Data collection from personal health devices.
- Data processing.
- User feedback provision in a closed-loop system.

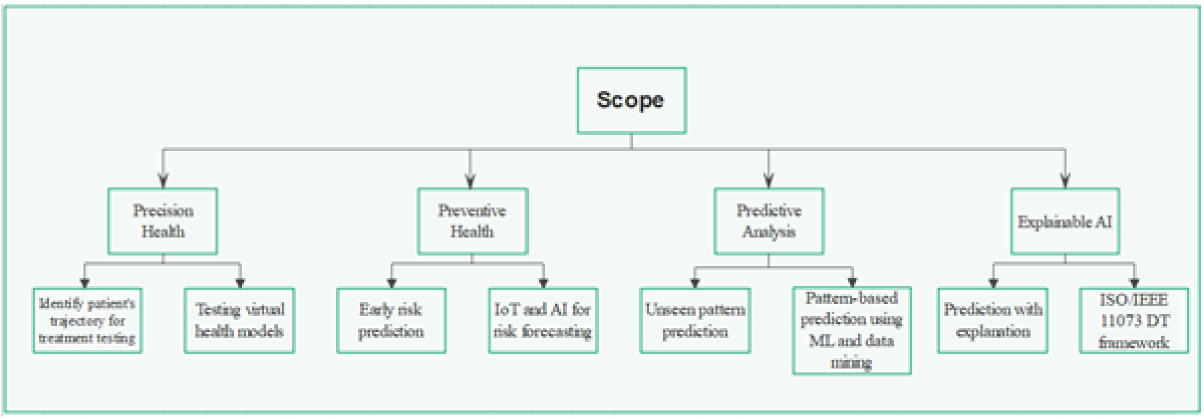


Figure 2.10. Overview of AI-integrated digital twin models for healthcare.

Overall, it is observed from Figure 2.10 that various health applications are supported by AI models in DT technology. To understand the types of AI models that support numerous healthcare applications, a discussion of the AI models proposed in DT literature for healthcare is presented. Based on a literature survey, AI models in DT for health are categorised by processing time and model type (Figure 2.11). The details are provided in the following points.

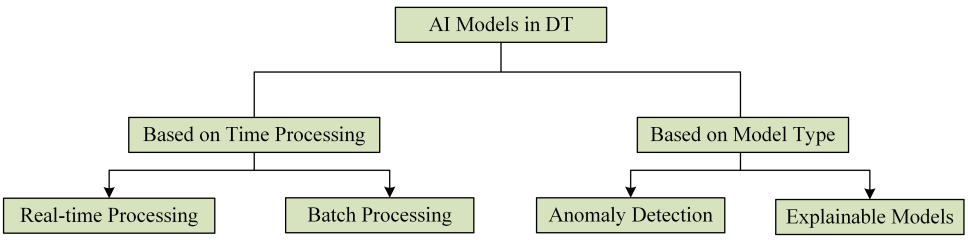


Figure 2.11. Types of AI models in DT for healthcare.

1. **Real-time Processing**
   DT systems are often applied in the visualization of data in real-time [121]. However, it is important to note that when real time prediction is incorporated, the definition of DT broadens greatly as follows [122]:

   - Extensive data is used to develop a predictive model.
   - The model continuously receives a stream of incoming data.
   - Predictions are generated by the model in real-time.

For example, a model trained with data on patient conditions and wait times can be used to forecast current waiting times in an emergency department. The model processes real-time data of incoming patients to predict their wait times. In [117], researchers created classifiers for real-time heart issue classification. They utilised CNN, LSTM, Support Vector Machine (SVM), and Logistic Regression (LR) to classify ECG heart rhythms in real-time. The DT model in [123] employed LSTM for predicting the vulnerability of lung cancer patients in real-time. Real-time prediction has also been examined for fitness management [121]. While the model in [117] concentrated on classifying data to detect health issues, the studies in [121] and [123] aimed at forecasting health outcomes before they occur.

2. **Batch Processing**
   Batch processing differs fundamentally from real-time processing, as it operates on static, periodic data [124]. An ML model is trained with a large dataset to predict outcomes for new test data. For example, a model trained on the activity data of individuals with obesity can be used to predict obesity in others. Batch processing models, built on extensive datasets, tend to be more precise than real-time models, which prioritize speed over accuracy [125].

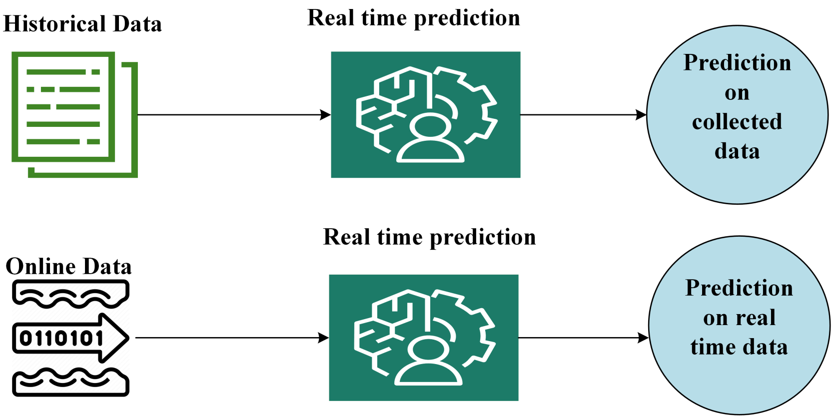


Figure 2.12. Techniques for real-time and batch processing.

Figure 2.12 illustrated that real-time processing uses live data, while batch processing relies on historical data. No specific classifications for batch predictions were identified in the study. The choice of learning techniques depends on the size of the training dataset. The following learning approaches for batch processing were found [126]:

- Batch Gradient Descent Learning: Applied when the batch size equals the entire training dataset.
- Stochastic Gradient Descent Learning: Used when the batch size is equivalent to a single data sample.

- Mini-Batch Gradient Descent Learning: Employed when the batch size is between one and the total number of samples in the dataset.

3. **Anomaly**
   Anomaly detection has been incorporated into DT models across various contexts. However, within healthcare, the concept of anomaly prediction has been proposed. The distinction between anomaly detection and anomaly prediction is subtle. As depicted in Figure 2.13, the detection process identifies anomalies in real-time data, whereas the prediction process relies on ML model built from historical data [126]. For instance, in [118], a DT model using CNN was applied for heart condition classification, while the approach in [127] focused on detecting anomalies in hospital data. Since anomaly prediction depends on ML models, it is important to be aware of the available approaches. Predictive DT models for healthcare are still in the early stages of adoption, resulting in limited examples in the literature. Therefore, both applied and potential techniques for anomaly prediction in DT models were considered. To train a supervised DT model for anomaly prediction, a labelled dataset distinguishing between anomalous and non-anomalous data is required. Neural Networks (NN) [121], SVM [128], K-Nearest Neighbour (KNN) [121], and Bayesian Networks (BN) [129] have been extensively proposed for these models.

   In contrast, unsupervised models do not require labelled data. These models assume that most data follow a normal pattern, with a small portion being anomalous. The model then classifies test data based on the similarity of its patterns to the presumed groups. Algorithms such as K-means, Autoencoders, Gaussian Mixture Modelling (GMM), and Principal Component Analysis (PCA) have been widely suggested for unsupervised anomaly prediction models [130].

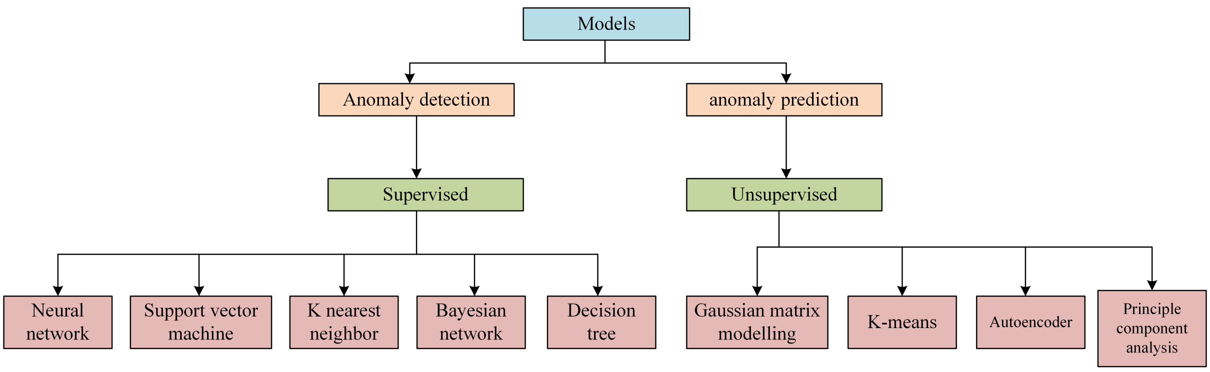


Figure 2.13. Distinguishing anomaly detection from anomaly prediction.

4. **Explainable Model**
   This type of model is a recent addition to DT predictive models. The challenge of trust in AI predictions has been widely discussed in the literature [131]. Explainable AI (XAI) was developed to enhance the interpretability and transparency of black-box

predictive models, like deep neural networks [132]. XAI models provide transparency by offering contextual explanations for their predictions [133]. Table 2.3 summarizes various prediction types applicable to Predictive DT models in healthcare. In one study [128], Lime algorithms were used to extract explanations for liver disease predictions. SVM classifiers were trained on liver disease detection data, and Lime algorithms identified the health parameters for patients diagnosed with liver disease. Another research [121] employed a counterfactual algorithm to recommend fitness precautions based on the user's activity history.

Table 2.3. Overview of DT Models in Healthcare

Ref.	Type	Requirements	Limitations	Healthcare applications
[128]	Explainable	Historical data	Still evolving, may not fit all classifiers (e.g., kernel-based SVM).	Personalized healthcare, preventive healthcare, participatory healthcare.
[134]	Anomaly	Historical data, labelled data for anomalies.	Implementation complexity	Preventive health (preparedness predictions)
[130]	Real-time	Live data.	Implementation complexity, reduced accuracy.	Predictive and personalized healthcare (preparedness predictions).
[121]	Batch	Large volume of historical data	Potential obsolescence, may not accommodate new data types.	Precision health and preventive health (diagnostic predictions).

In the realm of DT for healthcare, it has been observed that deep learning models using time-series data are predominantly employed for real-time processing. Additionally, supervised models utilizing historical data are commonly applied for diagnostic prediction. Emerging technologies such as XAI and transfer learning are being integrated into DT models. Numerous studies have proposed DT with AI capabilities for future healthcare applications. While prior work has explored the role of digital twins in real-time health monitoring and anomaly detection, existing frameworks often lack predictive capabilities, integration with real-time IoT data streams, and energy-efficient AI models. Moreover, most studies do not consider deployment constraints or patient-specific adaptation at scale. To address these challenges, the proposed research introduces cloud-based digital twin healthcare system that fuses patient monitoring with adaptive AI and task offloading, enabling scalable real-time prediction, low-latency communication, and improved system efficiency. This work contributes novel, deployable architecture that closes the gap between theoretical models and real-world digital health implementation.

2.5.5 Analysis of Model Techniques

Recent developments in asthma patient home monitoring involve DT systems, increasing reliability, efficacy, and capacity to evaluate breathing strategies and triggers and gather environmental data. The reliability of DT systems is indicative of their constant performance in the evaluation and interpretation of patient data, underscoring their dependability in practical medical contexts [135].

coring their dependability in practical medical contexts. DT aids in managing and treating asthma through mobile applications, gadgets, and remote monitoring systems, enabling early intervention, as described by Drummond et al. The diverse applications of DT technology in healthcare indicate its potential to enhance patient care and enable comprehensive monitoring [135].

Data acquisition in digital twin systems typically involves collecting real-time data from various IoT devices and sensors. This data is then standardised to ensure consistency and compatibility across different platforms and systems. Model evaluation techniques often include comparing the predictive accuracy of digital twin models with actual clinical outcomes, using metrics such as precision, recall, and F1-score.

2.6 Integration of IoT and Digital Twins

2.6.1 Role of IoT in Healthcare

The incorporation of IoT technology in healthcare has significantly transformed how health services are managed and delivered. IoT devices enable continuous patient monitoring, gather real-time health data, and provide critical information for prompt medical interventions. These devices include wearable sensors, smart medical devices, and remote monitoring systems, which are crucial in managing chronic diseases and preventive care [136].

Technological advancements and the rising demand for remote monitoring services have driven a substantial transformation in the healthcare industry. The expansion of digital healthcare applications has facilitated patient appointments and communication with physicians, regardless of geographic or time constraints, thereby enhancing healthcare accessibility and patient autonomy [137], [138]. Interconnected health devices are linked through centralized servers that manage health information, using both wireless and cable connections to ensure flexibility and reliability in diverse healthcare scenarios [139]. The functionality of Healthcare IoT (HIoT) networks heavily depends on their topology. The physical configuration of devices ensures compatibility for different use cases, typically including sensors, actuators, and workflows that communicate within the same application domain. This setup allows for simultaneous task performance and data recording, managed by service providers to ensure data security and privacy [140].

The prevalence of IoT-based smart wearable devices in personal healthcare is increasing. Devices such as smartwatches and fitness bands provide real-time health monitoring features, including heart rate, blood pressure, caloric burn, and sleep duration tracking. Data from these devices can help develop personalized health improvement plans. IoT-based software solutions enhance the functionality and usability of these devices, making them integral to modern healthcare systems [141]. By leveraging IoT capabilities, healthcare systems can address challenges related to data accessibility, patient monitoring, and real-time health management. Continuous advancements in IoT technology and its integration with healthcare systems hold significant promise for the future, enabling more effective and efficient patient care.

2.6.2 Synergies with Digital Twin Technology

Combining IoT with DT technology creates a robust framework for advanced healthcare solutions. DTs act as dynamic digital replicas of physical entities, enabling real-time interaction between virtual and physical worlds. The synergy between IoT and DT allows for enhanced monitoring, simulation, and prediction of health conditions. This integration supports the development of Personal Health Systems (PHS) and EHR for real-time patient monitoring [142]. The convergence of the IoT and digital twins is fundamentally transforming multiple sectors by creating a seamless connection between the physical and digital realms. This integration facilitates enhanced monitoring, analysis, and optimization capabilities. The IoT encompasses network of interconnected physical devices equipped with sensors, software, and network connectivity, enabling them to collect and exchange data. In contrast, digital twins are dynamic virtual representations of physical objects, systems, or processes that mirror real-world actions and characteristics [143], [144]. In manufacturing, digital twins enable operational analysis and improvement by creating virtual duplicates of manufacturing lines, which helps identify inefficiencies and bottlenecks without disrupting actual operations. IoT sensors continuously monitor equipment, allowing for predictive maintenance and reducing downtime. In healthcare, IoT devices gather extensive patient data, while digital twins simulate medical scenarios, aiding in research and personalised treatment plans. This technology enhances understanding of diseases, evaluates therapies, and trains healthcare professionals [145].

In smart cities, urban services such as traffic control and energy distribution are simulated and optimised through IoT and digital twins. Issues are predicted, different scenarios are experimented with, and resources are managed more effectively by urban planners using these technologies, leading to more sustainable and efficient urban living. In agriculture, data on soil, crops, and weather is collected by IoT sensors, while agricultural processes are simulated by digital twins to optimise yields and resource use, enabling data-driven decisions by farmers [146].

Supply chain efficiency is improved in logistics through the integration of IoT and digital twins, which track the real-time status of goods and simulate various scenarios to identify bottlenecks. In the energy sector, insights into energy consumption are provided by IoT sensors on equipment, while scenarios are simulated by digital twins to optimise energy efficiency and forecast performance. Educational experiences are enhanced through interactive simulations, and environmental conservation is aided by monitoring and modelling ecological impacts [147].

2.6.3 Integration of IoT and DT in healthcare

Several studies have demonstrated the potential of integrating IoT with DT in healthcare:

- Elderly Healthcare Services: Liu et al. explored innovative DT applications for elderly healthcare, highlighting the benefits of combining DT with big data, cloud computing, and IoT for enhanced service delivery [109].
- Personalized Therapies: Menon et al. provided a comprehensive analysis of virtual copies of patient anatomical structures for personalized therapies, showing the potential of DT in precision medicine [148].
- Cardiology and Lung Cancer Treatment: Corral-Acero et al. and Zhang et al. investigated the use of DT in cardiology and lung cancer treatment, respectively, demonstrating improved diagnostic accuracy and treatment outcomes through the integration of IoT and DT [123].
- Enhanced DT Construction Methodologies: Jia et al. explored methodologies for constructing enhanced DTs, focusing on the integration of IoT for real-time data collection and analysis [149].

2.6.4 Challenges in IoT and Digital Twin Integration for Healthcare

Despite the advancements, several challenges and gaps remain in the integration of IoT and DT in healthcare:

- Data Integration Complexities: Integrating data from various IoT devices into a cohesive DT model is complex and requires robust data standardization and interoperability frameworks [91].
- Privacy and Security: Another key concern is patient data privacy and security – a concern that becomes complex when patient health details collected through the IoT are involved [150].

- Real-Time Data Integration: Coordinating real-time data exchanged between IoT devices and DT models in a smooth manner is crucial for early detection of any problem requiring an action [151].

- Cost and Resource Constraints: As mentioned earlier, the complexities of evolving IoT and DT systems can be very costly and time consuming, thus their application in resource constrained environments may not be feasible [152].

Solving these problems calls for a systems approach by establishing workable and agile models, improving data protection mechanisms, as well as optimizing resource management. Future research should focus on overcoming these barriers to fully realize the potential of IoT and DT integration in healthcare.

2.7 Task Offloading Strategies in Healthcare

Task offloading is the process by which computational tasks are delegated from resource-constrained devices, such as smartphones or IoT sensors, to more powerful servers or cloud infrastructures. As depicted in Figure 2.14, this strategy is crucial for managing workloads efficiently, reducing latency, and conserving energy in devices that are not equipped to handle complex computations. In healthcare, the performance and reliability of applications requiring real-time processing and analysis of vast amounts of data can be significantly enhanced by task offloading [153].

A comprehensive overview of DT technology is provided by Md. Shezad Dihan et al., tracing its origins to NASA's Apollo missions and its evolution into a critical technology for various sectors. DTs are proposed as virtual representations of physical systems that facilitate intelligent decision-making through continuous bidirectional data flow between physical and virtual entities. The applications of DTs in manufacturing, urbanization, agriculture, medicine, robotics, and military/aviation are explored, with emphasis on distinct data collection, storage, processing, and analysis techniques in each field as shown in Figure 2.15. Current challenges and future directions are identified, particularly the need for advanced modelling strategies, real-time data processing, and enhanced data security. Through detailed comparative analysis, diverse sector-specific insights are unified, and pathways for leveraging DT technology to optimize system performance, resource management, and decision-making across various domains are proposed [154].

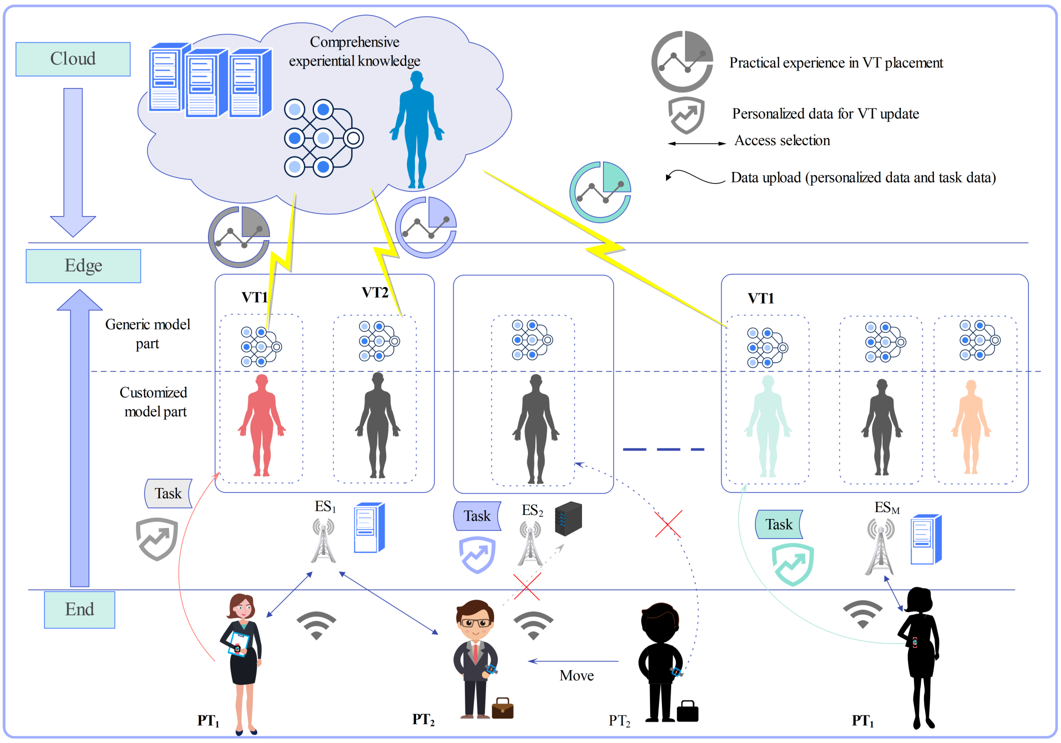


Figure 2.14. **The end-edge-cloud collaborative HDT system [153].**

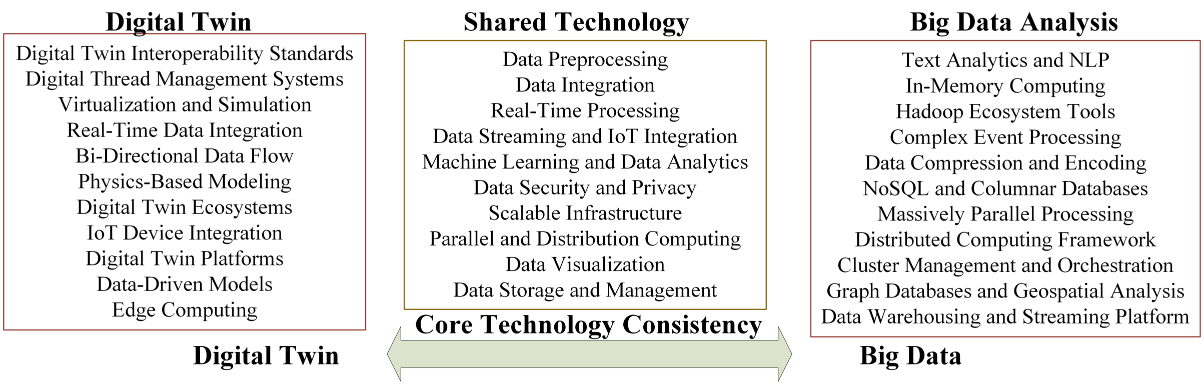


Figure 2.15. **Analytical consistency between digital twins and big data [154].**

2.7.1 Importance in Healthcare Context

The integration of task offloading in healthcare is of paramount importance due to several reasons:

- Resource Efficiency: Healthcare applications, such as telemedicine, patient monitoring, and diagnostic systems, often operate on devices with limited computational power. Offloading tasks to more capable infrastructure ensures these applications run smoothly without overburdening the local devices.
- Real-time Processing: Many healthcare services rely on real-time data processing and

analysis. For instance, continuous patient monitoring systems must analyse vital signs promptly to detect any anomalies. Task offloading to cloud or edge servers facilitates the rapid processing of such data, leading to timely and potentially life-saving interventions.

- Data Security and Privacy: Task offloading can also enhance data security and privacy. Sensitive health data can be processed and stored in secure cloud environments with robust encryption and compliance with healthcare regulations, such as HIPAA.
- Scalability: As the volume of healthcare data grows, scalable solutions are required to manage and analyse this information effectively. Cloud-based task offloading offers the flexibility to scale computational resources according to demand, ensuring consistent performance and reliability.

2.7.2 Review of Existing Algorithms and Strategies

In healthcare, there is an increasing emphasis on the necessity for efficient data processing and effective energy management in medical devices, which has attracted significant academic interest. Research in this domain spans task offloading, Digital Twin technologies, and the integration of social determinants in health informatics. Scholarly discussions acknowledge considerable advancements in these areas while also identifying persisting gaps [155].

In recent research by Jeremiah et al., DT-assisted vehicular edge computing was examined to enhance network services through edge collaboration and precise resource allocation. The study confirmed the feasibility of using non-orthogonal multiple access and dynamic roadside unit selection based on channel state information. Furthermore, the research explored managing complex optimization tasks such as task offloading, decision-making, subchannel assignment, and RSU connections using advanced high-level policy gradient algorithms like the Advantage Actor-Critic algorithm. Simultaneously, Qiu et al. are investigating a DT-assisted edge computing offloading approach utilizing IBMPA to efficiently and swiftly utilize available energy and computational resources within a stringent time frame [156].

Building on the discussion, Bozkaya et al. proposed a task computation offloading scheme for DT-enabled networks (DTEN) that is mindful of both energy consumption and delay [157]. However, Chen et al. assess a computation offloading and service caching approach for DTEN that used A3C algorithm along with dependency features. From their studies, they concluded improved energy efficiency and system performance [158].

Table 2.4, shown overview of various features and offloading strategies used in different works. In This Work, the DTH-ATB-MAPPO offloading strategy was employed, with a strong focus on the integration of DT. Cybersecurity was addressed through the ACTO Algorithm, and multiple communication protocols were supported. Significant improvements were observed in energy efficiency and network latency with 30 MEC nodes, achieving 53.8% and 33.4% reductions, respectively. Predictive healthcare interventions were supported, and social health determinants were included, with extensive real-time testing conducted. In the 2024 study [153], the TACO offloading strategy was used, also integrating DT and emphasizing cybersecurity via blockchain technology. Supported communication protocols included D2D, D2C, and C2E. Energy efficiency was improved by 25%, with a corresponding 25% reduction in network latency. However, predictive healthcare interventions and social health determinants were not extensively covered, and real-time testing was not as thorough.

The 2022 study [159] employed the DTTOS offloading strategy, integrating DT and focusing on cybersecurity with blockchain. Supported communication protocols included D2D, V2I, and D2C. Energy efficiency was improved by 60%, with 24.39% reduction in network latency. Predictive healthcare interventions were supported, but social health determinants were not included. Extensive real-time testing was performed. Finally, the 2024 study [160] utilized the OWS offloading strategy with DT integration but lacked a specific cybersecurity focus and supported communication protocols.

This comparative analysis shows that the approaches and focal points of the various studies differ in terms of offloading techniques, DT integration, security, communications, power consumption, and latency as well as the integration of social health determinants and real-time testing.

Table 2.4. Overview of Features and Offloading Strategies in Different Works

Feature	2024 [3]	2024 [153]	2022 [159]	2024 [160]
Offloading Strategy	DTH-ATB-MAPPO	TACO	DTTOS	OWS
Integration of DT	Yes	Yes	Yes	Yes
Cybersecurity Focus	ACTO Algorithm	Blockchain	Blockchain	No application
Communication Protocols Supported	Multi-protocol	D2D, D2C, C2E	D2D, V2I, D2C	No application
Energy Efficiency Improvement	53.8% with 30 MEC nodes	25%	60%	N/A
Network Latency Reduction	33.4% with 30 MEC nodes	25%	24.39%	N/A
Predictive Healthcare Interventions	Supported	N/A	Supported	Suggest support
Inclusion of Social Health Determinants	Yes	No	No	No
Real-Time Testing	Extensive	Not extensive	Extensive	N/A

2.7.3 Addressing Gaps in Healthcare Task Offloading

Despite these, a number of challenges and gaps are still present regarding diverse task offloading solutions that have been proved to be useful in different scenarios and these include:

1. Adaptability and Scalability: Most of the current methods do not possess enough adaptability to constantly changing network characteristics, computational demands, and patients' requirements.
2. Integration with Digital Twins and Social Health Determinants: Digital Twins and the concept of social health determinants are two new ideas in this context that come as opportunities and risks in the offloading of tasks in real time. To achieve real-time synchronization of the digital twins with the offloaded task, there must be a proper coordination in the flow of information. In the same way, integration of social health determinants into offloading plans can enhance the applicability of the healthcare services.
3. Security and Privacy Concerns: Security and privacy of health data while offloading is still a big concern among all the challenges.
4. Energy Efficiency: Despite the fact that offloading may reduce power consumption in local devices, the total energy chain including the cloud servers has to be minimized. Finding a proper solution for the problem of achieving a good balance between computational capabilities and energy concern is not an easy one.

Thus, a comprehensive task offloading framework has been proposed to meet the lack of the reviewed literature in the healthcare application domain. This framework switches between partial and full offloading while also incorporating the digital twin and social health determinant considerations into the process. Also, the ACTO algorithm is used to detect threats and counter them using suitable cybersecurity measures. Through meticulous evaluation, the framework's superior efficacy in augmenting computational efficiency, conserving energy, and elevating patient outcomes has been delineated, marking a significant advancement in harmonizing technology with healthcare. Although recent studies have introduced heuristic and learning-based offloading strategies, they typically lack adaptability to changing healthcare conditions, overlook the integration of digital twins and social health determinants, and provide limited cybersecurity measures. Additionally, few models have demonstrated real-time applicability in clinical settings. To address these shortcomings, the proposed research presents a hybrid task offloading model incorporating both partial and binary offloading schemes. It integrates real-time digital twin synchronization, considers social health indicators for prioritization, and embeds a lightweight cybersecurity module using the ACTO algorithm. This approach fills critical gaps in the literature by enabling dynamic, secure, and patient-centric task management for next-generation healthcare systems.

2.8 Comparative Analysis of Existing Work

1. **CNN Architectures on FPGAs:**

   - Implementation and Performance: CNNs have been successfully implemented on FPGAs, enhancing medical diagnostics through efficient processing of ECG, EEG, and EMG signals. Examples include Jiahao et al.'s FPGA design for ECG classification, achieving high processing rates and resource efficiency, and Guo et al.'s Angel-Eye architecture, which improves speed and energy efficiency through data quantization techniques.
   - Challenges and Solutions: Key challenges such as scalability, resource utilization, and integration with existing systems were addressed through various FPGA accelerator designs, highlighting the balance between speed, accuracy, and power consumption.

2. **Cloud-Based Digital Twin Ecosystems:**

   - Applications and Benefits: DTs provide real-time monitoring, predictive analytics, and personalized treatment plans in healthcare. Studies have shown their effectiveness in chronic disease management, real-time health monitoring, and improving diagnostic accuracy.
   - Challenges: The primary challenges include data standardization, real-time synchronization, and integration with IoT devices. Researchers have proposed solutions like the Lung-DT framework and the Virtual Human Twin to enhance the practical application of DTs in healthcare.

3. **Task Offloading Strategies:**

   - Importance and Effectiveness: Task offloading enhances computational efficiency, real-time data processing, and energy conservation in healthcare applications. Strategies integrating DTs and social health determinants have shown potential in improving patient outcomes and system performance.
   - Emerging Solutions: Innovative approaches like the ACTO algorithm for adaptive cybersecurity mechanisms and heuristic greedy and DQN-based strategies demonstrate advancements in task offloading methodologies.

2.8.1 Comparative Analysis

The comparative analysis of methodologies, results, and conclusions reveals significant advancements in healthcare technology. CNN architectures on FPGAs utilize parallel processing and data quantization to optimize performance, while cloud-based DT frameworks rely on IoT integration and real-time data analytics. Task offloading strategies employ both heuristic and machine learning algorithms to manage computational loads and enhance security. The results indicate that CNN-FPGA implementations have significantly improved processing speeds and diagnostic accuracy in medical applications. Digital twins have advanced real-time health monitoring and personalized treatment, although integration challenges remain. Task offloading strategies have demonstrated improvements in computational efficiency and energy conservation yet require further development for broader applicability. The reviewed works collectively highlight the need for adaptable, secure, and energy-efficient solutions in healthcare technology. The integration of DTs and social health determinants into task offloading strategies presents new research opportunities to enhance healthcare delivery. To sum up, the comparative analysis reveals that healthcare technology has advanced tremendously, but there is still more research that can be conducted and developed to increase the impact and field of use of such technologies.

2.9 Positioning of Current Research

The current work expands on extant literature by incorporating updates on CNN architectures implemented on FPGAs, the establishment of cloud-based digital twin environments, and the tasks offloading mechanisms. These integrations are designed in order to solve quite vast gaps in the area of healthcare technologies. This case aims at improving computational speed, real time data processing and personalization, which are essential in modern healthcare applications.

The formulation of an adaptive task offloading system that integrates between partial and full offloading. This framework introduces concept of a digital twin and social health determinants into decision making where treatment is more accurate and suited to the need. Additionally, due to the application to medical devices, an energy efficient algorithm has also been incorporated to enhance the performance of the framework. Also, the opportunity to use the ACTO algorithm was aimed at implementing an effective response to cybersecurity threats in a nontechnical environment dependent on the situation, which also enhances the effectiveness of cybersecurity measures.

2.10 Summary and Conclusion

In this chapter, a review of the literature pertaining to the research areas of the study was presented, including design and implementation of efficient CNN architectures for health-care devices using FPGAs, the creation of cloud-based digital twin environments for health-care delivery, and novel task offloading mechanisms in health-care applications. Basic notions and definitions were introduced and defined to build the proper framework of the conversation. The chapter then delved into specific areas of related work, examining CNN architectures on FPGAs, cloud-based digital twin ecosystems, task offloading strategies in healthcare, and the integration of IoT and digital twins. Through a systematic review of relevant studies and frameworks, critical gaps and limitations in the existing body of knowledge were identified.

Chapter 3

Efficient CNN Architecture on FPGA Using High Level Module for Healthcare Devices

3.1 Introduction

This chapter perfectly examines the feature used to create an effective one -dimensional convolutional neural network (1-D CNN) architecture, which has been used on field programmable gate array (FPGA) platform for wearable healthcare equipment. These modern health tools require high performance, low energy consumption and accurate diagnosis, especially when handling complex biosignals such as ECG, EEG and EMG. The main goal of research is to design, implement and evaluate the 1-D CNN architecture that meets these demand criteria.

The CNN model was used due to the benefits of raw data learning facilities, which is important in biomedical signal therapy. Using this model on the FPGA platform takes into account parallel calculations of FPGAs, which improves the computation rate and power use. FPGA is also used flexible and reconstructive, as used here in wearable healthcare devices, which has dynamic specifications.

FPGAs were chosen in this research, as they were due to parallel processing, low latency and energy efficiency execution to provide important properties for portable health equipment, which require signal classification in real time during power and space barriers. Unlike general-purpose CPUs or GPUs, FPGAs offer hardware-level customization, enabling tailored acceleration of convolutional operations without incurring the overhead of unnecessary components. Their reconfigurability makes them ideal for prototyping evolving healthcare models while maintaining consistent performance across diverse biosignal types such as ECG, EEG, and EMG. This ensures responsive and accurate health monitoring in resource-constrained edge environments. In this study, throughput is reported in Giga Operations per Second (GOPS) instead of floating-point operations, as fixed-point arithmetic was adopted for efficient FPGA implementation.

This chapter is organized as follows: Section 3.2 outlines the method used in the study and Section 3.3 describes the design of the 1-D CNN accelerator as presented in this research work. In Section 3.4 describes the tools and instruments that were utilised in this contribution. The details of data collection and preprocessing methods are discussed in Section 3.5. In Section 3.6 describes the approach that will be used in the analysis of the collected data. Furthermore, the difficulties and the limitations encountered in the study are discussed in Section 3.7. In the last Section 3.8, conclusions and summaries are presented.

3.2 The Proposed 1-D CNN Accelerator

This section presents the methodology used to design and evaluate 1-D CNN architecture for classifying ExG signals, namely ECG, EEG, and EMG on FPGA platform. The objective was to meet the high-performance, low-latency, and energy-efficiency demands of wearable healthcare devices. A structured experimental framework was adopted, as summarized in Table 3.1, encompassing data collection, model development, training and validation, and hardware implementation.

Table 3.1. Key stages of research methodology.

Stage	Description
Data Collection	Gathering ECG, EEG, and EMG signals from various datasets.
Preprocessing	Applying padding, reshaping, and resampling to standardize data.
Model Development	Designing 1-D CNN architecture with convolutional, pooling, and fully connected layers.
Training and Validation	Using training data to optimize model parameters and validation data to tune hyperparameters.
Hardware Implementation	Implementing the trained model on Xilinx Zynq xc7z045 FPGA, optimizing for parallel processing and low power usage.
Performance Evaluation	Assessing model using metrics like accuracy, precision, recall, F1-score, and energy efficiency.

NN advancements have improved pattern classification and data mining studies. Recently, many machine learning tasks that heavily relied on handcrafted feature engineering have been transformed by end-to-end deep learning models, such as CNN [55]. Numerous computational layers, organized as directed acyclic graphs, form a 1-D convolutional (CONV) layer. Each layer abstracts the data from the preceding layer into a feature map. The result y_n is expressed as:

$$y_n = b_n + \sum_{k=0}^{K-1} w_{nk} x_k, \tag{3.1}$$

where x_k represents the input feature map data. The output Y_k after applying the activation function f, commonly a rectified linear unit (ReLU), is:

$$Y_k = f(y_n). \tag{3.2}$$

CNNs were selected for their strong feature learning capabilities from raw biomedical signals, eliminating the need for manual feature extraction. A 1-D CNN architecture was adopted due to its suitability for time-series ExG signals. The core layers included convolutional, pooling, and fully connected layers, culminating in a SoftMax classifier for final output. The ReLU activation function was used throughout the network:

$$\mathrm{ReLU}(x) = \max(0, x). \tag{3.3}$$

Moreover, FPGA acceleration for CNNs has attracted considerable attention. An FPGA accelerator that is well-designed for CNN can fully exploit parallelism, achieving low latency and high speed, meeting the demands for high performance, high speed, and low power consumption in various applications. FPGAs are widely utilized as economical solutions across many industries. Additionally, the reconfigurability of FPGAs allows them to swiftly adapt to new CNN designs [53]. FPGAs offer better energy efficiency compared to CPUs or GPUs. Designing a high-performance FPGA accelerator is nontrivial and entails several steps such as parallel architecture exploration, memory bandwidth optimization, area-timing trade-offs, and access-handling interfacing with software. As a result, automatic compilers for FPGA-CNN accelerators have been proposed to automatically generate hardware descriptions of targeted accelerators based on parametric templates and provide an integrated design space that co-optimizes network structure-related parameters with respect to target deployment platforms as well as schedule parallel optimizations [59]. In this study, lightweight 1-D CNN architecture was optimized for FPGA deployment, using shift-based arithmetic instead of multipliers, pipelined processing elements, and memory-aware scheduling. These choices contributed to improved speed, reduced power consumption, and effective resource usage, forming the foundation of the proposed FPGA-based ExG classifier.

3.2.1 Signal Flow Graph and Processing Element Design

A signal flow graph is a methodology to represent discrete time systems, which displays the sequential processing of data or enumeration. In the following, this method is used to illustrate the 1-D CNN CONV layer (also known as computation) and the processing element (PE) design in iterations of data processing and classification, as shown in Figure 3.1. The output is obtained after multiplication and addition operations in the 1-D CNN CONV layer. This method uses variables w_{nk} and h_{ki}, representing the concatenation of kernel weights and the feature map of the input data, respectively, along with a bias b_n. To minimise the hardware requirements for the 1-D CNN, the multiplication operation is performed with a left shift. Continuous collection operations follow, requiring a register (R). Once the iterative data processing cycle is completed in R, it is then combined with b_n. Ultimately, the final output y_k is produced.

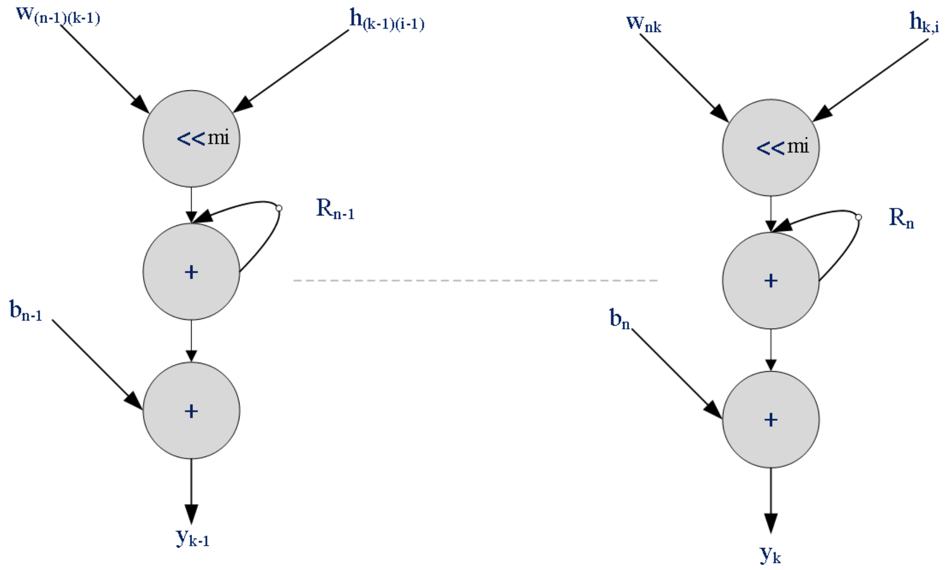


Figure 3.1. Schematic of a 1-D CNN CONV layer's signal flow.

To reduce the hardware complexity, multiplication operations were replaced by shift operations, as shown in Figure 3.2. The equation for the shift operation is given by:

$$X_k = h_{k,0} + h_{k,1} \cdot 2^m + h_{k,2} \cdot 2^{2m} + \ldots + h_{k,i} \cdot 2^{im}. \tag{3.4}$$

Where i is the partition of kernel weights, and m is the length of the shift. The substitution of the shift operation into the convolution equation:

$$X_k = h_{k,i} \cdot 2^{im}, \tag{3.5}$$

where i^{th} represents the partition of the kernel weights of the 1-D CNN, with the shift length $m = \frac{k}{i}$. Subsequently, Eq. (3.5) is substituted into Eq. (3.1), resulting in the following equation:

$$y_n = b_n + \sum_{k=0}^{K-1} \sum_{i=0}^{i-1} h_{k,i} \cdot 2^{im} \cdot w_{nk}. \tag{3.6}$$

The final form of the proposed architecture is defined by Eq. (3.6). As shown in Figure 3.2, the PE includes an XOR gate that functions as a selector to check the sign bit. To reduce the number of components required, a tristate buffer is used in place of other gates within the multiplexer.

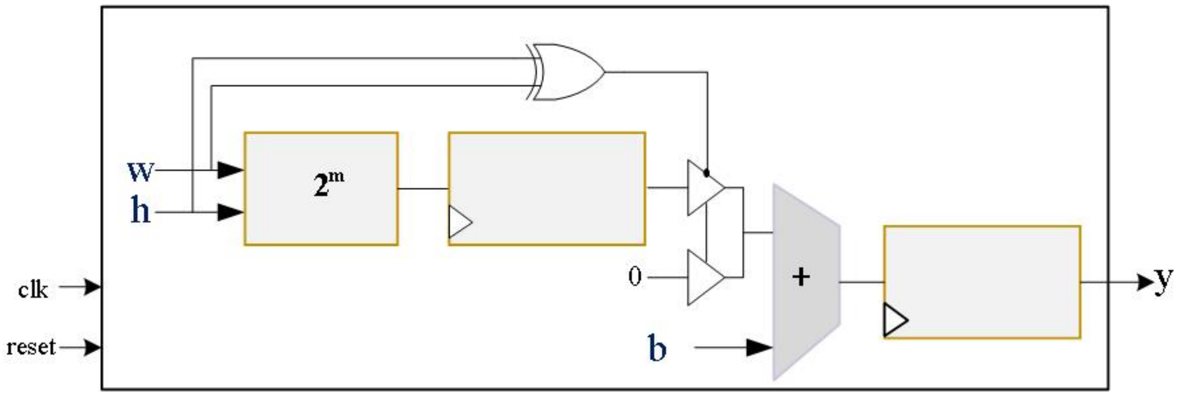


Figure 3.2. Configuration of a processing element (PE).

3.2.2 Theoretical Compute Peak Performance

The theoretical compute peak performance of an FPGA can be determined using the following equation:

$$P = f_{\text{max}} \times \left(\frac{R_{\text{total}}}{R_{\text{unit}}}\right) \tag{3.7}$$

This equation estimates the upper limit of the FPGA's computational capability based on its hardware resources and operating frequency, as described in [115]. The equation components are:

- P: Theoretical peak performance in operations per second (op/s), i.e. the maximum number of operations FPGA can perform under ideal conditions (ops/second).
- f_{max}: It is the highest clock frequency that can be achieved from a single operation core. The f_{max} is the maximum frequency at which the FPGA can run its cores, usually described in megahertz (MHz) or gigahertz (GHz). Higher frequencies enable quicker operation executions.
- R_{total}: The number of total hardware resources available for that type on FPGA. These resources include Look-Up Tables (LUTs), Flip-Flops (FFs), and Digital Signal Processors (DSPs), each having a finite quantity available for implementing computational cores.

- R_{unit}: Signifies the hardware resources required to implement one operation core of the desired computation. This includes the specific amount of LUTs, FFs, DSPs, or other resources needed to create a functional unit capable of performing the operation.

The peak performance P is calculated by multiplying the maximum frequency f_{max} at which the operation cores can run by the number of such cores that can be implemented on the FPGA, given its available resources. The equation ensures that the estimation is constrained by the most limiting resource, providing a realistic upper bound for the FPGA's performance.

3.2.3 Hardware Implementation

The trained 1-D CNN model was deployed on the Xilinx Zynq xc7z045 FPGA platform. Due to the substantial size of the kernels and weights, all parameters were stored in off-chip memory. However, as on-chip buffers were insufficient to cache the entire network, a hybrid memory strategy was adopted: data and weights were preloaded from off-chip memory into an on-chip buffer to facilitate high-speed access during computation, as illustrated in Figure 3.3.

This buffering scheme enabled efficient feeding of data into the processing element (PE) arrays via a data bus, maximizing memory bandwidth and supporting concurrent I/O and computation. To further optimize throughput, data was loaded in batches rather than individually. Since a single output channel may require multiple compute cycles, intermediate results were passed to an output buffer before being written back to off-chip memory.

The logic flow of the acceleration process is described in Algorithm 3.1, which initializes memory, distributes data and weights, computes convolutions in parallel using PEs, applies activation and pooling, and stores the results. The accelerator pipeline includes specialized functions for memory loading/storing, convolution via PE, adder tree reduction, ReLU activation, and max pooling.

Algorithm 3.1. 1-D Convolutional Neural Network (CNN) Accelerator

```
Begin 1-D CNN Accelerator                                   # Initialize Off-chip memory
Initialize off chip memory           # Load data from Off-chip memory to On-chip buffer.
data = Load(off chip memory, data bus)
Store(on chip buffer, data)                                 # Load weights to Weight buffer
weights = Load(off chip memory, data bus)
Store(weight buffer, weights)                               # Initialize Input Signal Buffer
input signals = Load(input signal buffer)           # Initialize Processing Elements (PEs)
for each PE in PE array do
    Initialize PE
end for                                          # Data and Weight distribution to PEs
for each input signal in input signals do
    for each weight in weights do
        Distribute input signal to PE, Distribute weight to PE, output PE = PE(input signal, weight), Collect
outputs from PE to adder tree.
    end for
end for
sum = Adder_Tree(output_PE), relu output = ReLU(sum), max pooled output = Max_Pooling(relu output) #
Adder Tree Processing, Activation function (ReLU).
Store(output buffer, max pooled output)                # Write the processed to Output Buffer
Write(off chip memory, output buffer)            # Transfer Output Buffer to Off-chip memory
End 1-D CNN Accelerator
function LOAD(memory, bus)
    data = Read from(memory, bus)
    return data
end function
function STORE(buffer, data)
    Write to(buffer, data)
end function
function PE(input signal, weight)
    result = input signal * weight                 # Perform convolution or other operations
    return result
end function
function ADDER TREE(outputs)
    sum = 0
    for each output in outputs do
        sum += output
    end for
    return sum
end function
function RELU(value)
    return max(0, value)
end function
function MAX POOLING(inputs)
    max value = −∞
    for each input in inputs do
        if input > max value then
            max value = input, then Write to(memory, data)
        end if
    end for
    return max value
end function
```

This process continues with the initialization of the input signal buffer, which holds incoming signals prior to processing. Each processing element (PE) in the PE array is then initialized to perform parallel operations.

During execution, input signals and their corresponding weights are streamed to the PEs. Each PE performs a convolution operation and generates partial outputs. These outputs are aggregated using an adder tree, and the result is passed through a ReLU activation function to introduce non-linearity.

Following activation, the data undergoes max pooling, a downsampling step that preserves essential features while reducing dimensionality. The pooled outputs are stored in an output buffer, which is then written back to off-chip memory, completing the dataflow pipeline. The overall process is facilitated by modular functions:

1. Load retrieves data from memory.
2. Store writes data to internal buffers.
3. PE performs core multiply-accumulate operations.
4. Adder Tree sums PE outputs.
5. Max Pooling extracts dominant features.
6. Write transfers final outputs to memory.

This streamlined accelerator architecture enables efficient pipelining of computation and memory operations, leveraging FPGA parallelism to achieve high-throughput, low-latency processing suitable for real-time biomedical applications.

3.2.4 Model Development

Introduction of 1-D CNN architecture comprising convolutional layer, pooling layer, and fully-connected (FC) layers. The architecture is optimized to minimize the amount of resources used while providing sufficient flexibility for mapping designs onto FPGA. The proposed CNN accelerator architecture is illustrated in Figure 3.3.

Key features of the model development included:

- Pipelined processing unit array application for high-capacity and efficient operation.
- Implementation of shift operations to reduce hardware complexity.
- Improvement in consuming less resource by integrating a tristate buffer in the multiplexer.

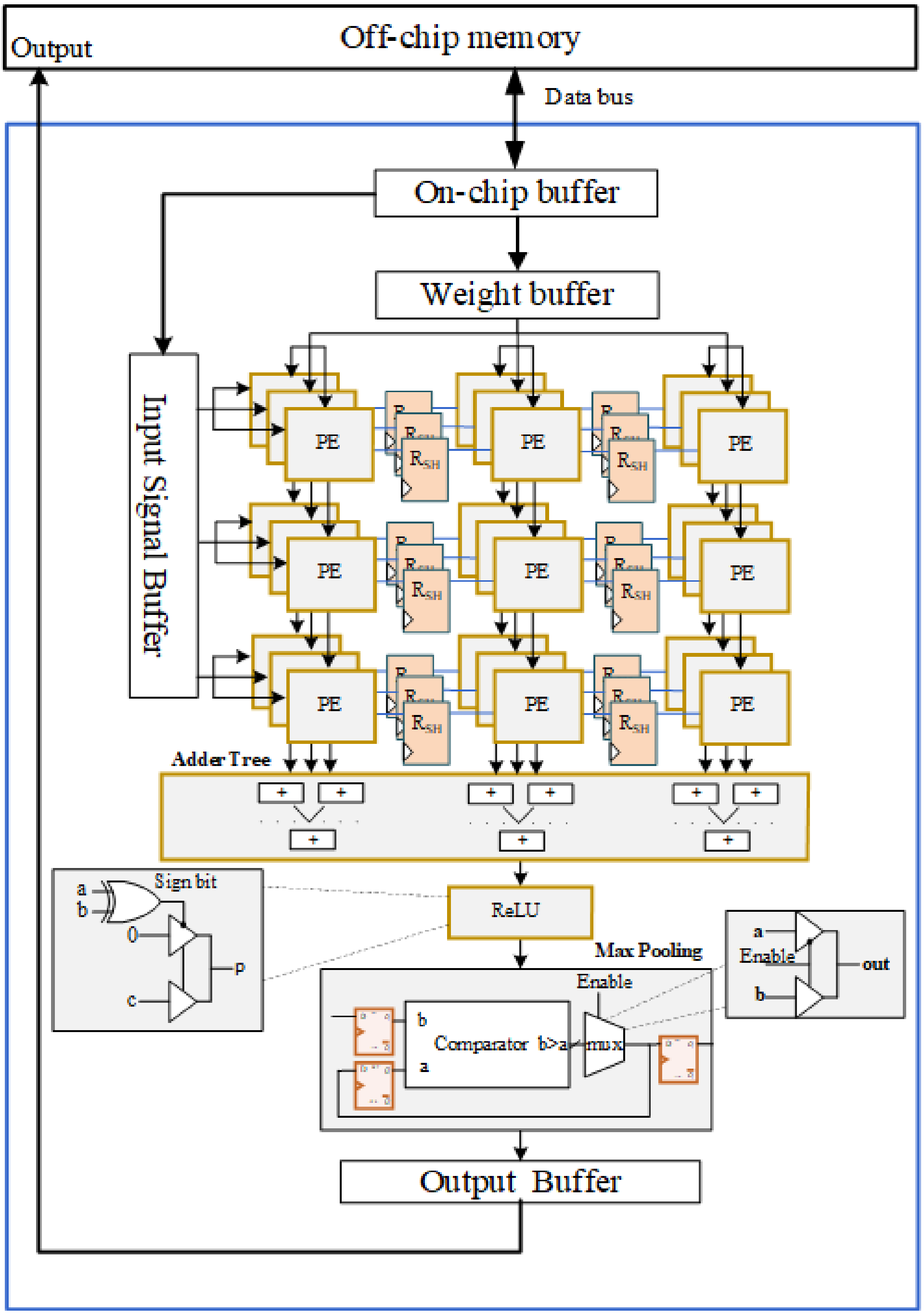


Figure 3.3. **Layout of the proposed CNN accelerator.**

3.3 Proposed 1-D CNN Structure and Other Algorithms

3.3.1 System Workflow

The workflow during the system building phase is shown in Figure 3.4. After collecting the dataset, it is stored in a database for easy fetching and analyzing. The stored data is then passed for preprocessing, which includes padding, reshaping, and resampling. The data is also split into two parts called testing data and training data, which will be used in the model-building process. The model creation phase has two steps: 1) Model Training and 2) Model Evaluation. In the evaluation phase, the testing data is used to test how well the model is working or if there may be an underfitting problem with the training set. The ensemble is then utilized on the three models to aggregate the insights and output, identifying which mode performs best.

This step is repeated 10 times to select the best model from these ML algorithms. After storing the best model, the system is ready to accept any sample data for classification. Finally, the algorithm is implemented on FPGAs to create an accelerator for classification.

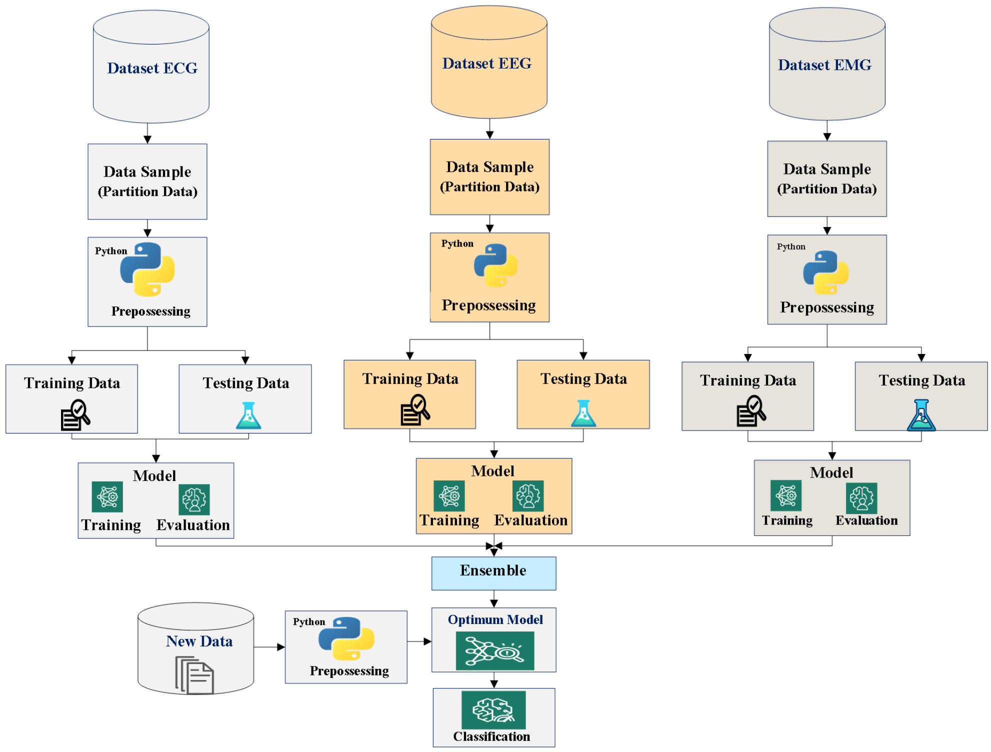
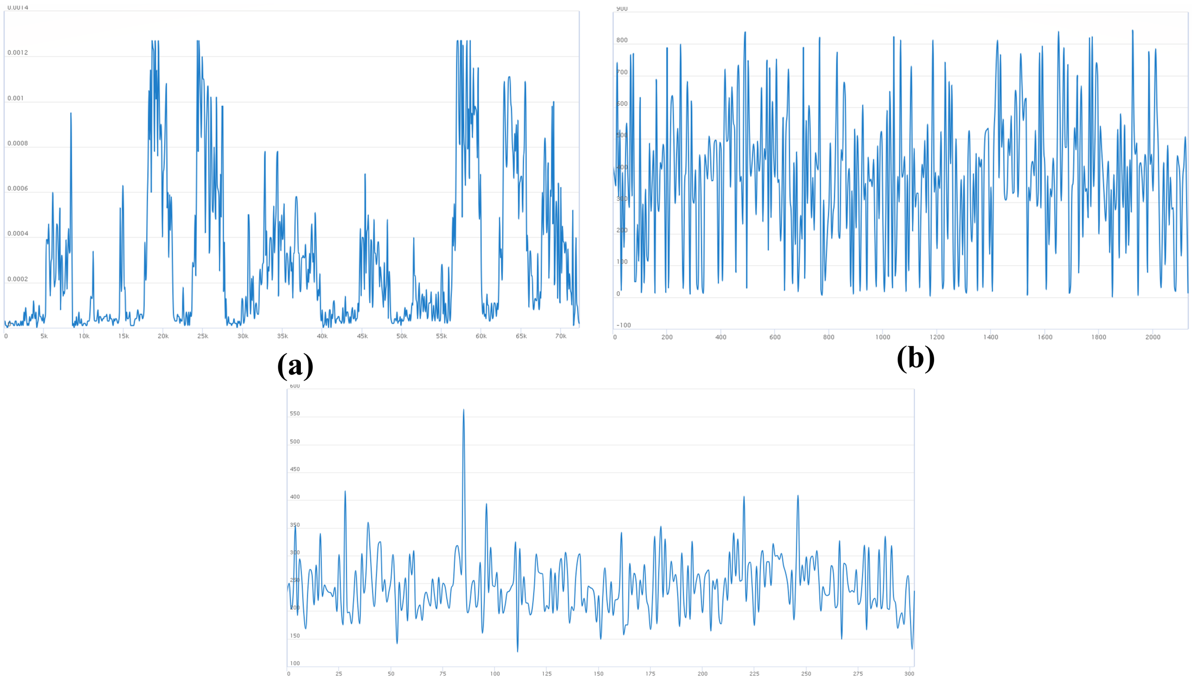


Figure 3.4. Proposed framework and workflow for ExG signal classification.

3.3.2 Data Utilization

In this chapter, ExG signals, including ECG, EEG, and EMG signals, were used due to their similar characteristics, as depicted in Figure 3.5. Summary of the ECG signal data set [67] from the UCI machine learning repository. There are several parameters in the data set, such as a target state of heart disease or not having heart disease. Although the ECG database contains only 303 patients, 4242 parameters, and 76 features, published research typically leverages merely up to a few dozen related measurements (14 in our case). The sampling frequency of the ECG signal was set to 100 Hz in this work. The dataset of the ECG signal includes: age (years); sex (1= male, 0 = female); type of chest pain; resting blood pressure; serum cholesterol level; fasting blood sugar > 120 mg/dl (1=true, 0=false); resting electrocardiographic results; maximum heart rate achieved; exercise-induced angina (1=yes, 0=no); ST depression induced by rest relative to exercise; normal, a fixed defect, and a reversible defect are indicated with the numbers 3, 6, and 7, respectively; target (disease) (1=yes, 0=no).



(a) (b) (c)

Figure 3.5. ExG signals include: a) EMG, b) EEG, c) ECG.

The brainwave data set is processed for the EEG signals in [76], [83]. It used dry electrodes to delineate the positive, negative, or neutral state at each step experienced by participants. In this study, the types of sentiments were divided into three categories based on a 1/2/3 coding scheme: melancholy/negative (coded as "1"), joyful/positive (coded as "2"), and neutral (coded as "3"). The EEG signal has a sample size of 1492 number of instances, 2548 features, and the number of parameters is 3801616 with a sampling frequency of 150Hz.

The EMG signal dataset from the UCI Machine Learning Repository was employed [45]. This dataset tracked the movements of the volunteers and normal as well as aggressive actions that were performed along with their impact in real time by attaching eight channels on the body. The EMG signal dataset consists of the total of 10000 instances, 8 features, and 723220 parameters with the sampling frequency of 200 Hz. For simplicity of classification, this sheet of EMG signal data was encoded from 0 to 6, with channel number 7 for the data carrier sensor.

3.4 Materials

This section describes the hardware and software tools employed in the study to design and build the 1-D CNN accelerator.

3.4.1 Hardware

The proposed 1-D CNN architecture was implemented on the Xilinx Zynq XC7Z045 FPGA platform, selected for its high parallel processing capabilities, low latency, and energy-efficient operation key requirements for real-time signal classification in wearable healthcare systems.

- **FPGA Platform:** Xilinx Zynq XC7Z045 combines programmable logic with ARM-based processing, which offers a balanced architecture that is suitable for distributing deep learning models with customized resource control. DSP slices with high numbers, blocking RAM and logic cells make it ideal to accelerate fixed operations in 1-D CNN, while at the same time the power consumption minimal necessary criterion in edge-based medical monitoring.

The platform enabled the integration of all shift-based multiplier and light buffers such as pipe CNN calculation, real-time data flow control and hardware-friendly optimization in a compact and reconstruction environment.

3.4.2 Software

The development, training and evaluation of the 1-D CNN model was done using the Python programming language, which was chosen for its widespread support in scientific data processing and machine learning. Jupyter Notebook was used as a primary development environment, enabling effective visualization, troubleshooting and iteration during model development.

Many python libraries were used throughout the process. Numpy and pandas were employed for numerical operations and preprocessing tasks, such as reshape, padding and generalization of ECG, EEG and EMG Signal dataset. The performance of the model was assessed by the use of Scikit-learn, which provides metrics as accuracy, precision, recall, F1 score and AUC.

In addition to the proposed 1-D CNN model, several classic machine learning algorithms were used on performance comparison:

1. Support Vector Machine (SVM) with a radial basis function (RBF) kernel and regularization parameter $C = 1.0$.
2. Naïve Bayes (NB) using Gaussian assumptions for feature distributions.
3. Logistic Regression (LR) with L2 regularization and solver = 'liblinear'.
4. K-Nearest Neighbors (KNN) with $k = 5$.
5. Decision Tree (DT) with maximum depth = 10 to prevent overfitting.
6. Random Forest (RF) with 100 estimators and bootstrap sampling.
7. Stochastic Gradient Descent (SGD) classifier using hinge loss and learning rate = 0.01.

All models were trained using the same preprocessed data and evaluated using a 5-fold cross-validation to ensure fairness. This comparative assessment helped show the performance gains to the 1-D CNN model, especially in the context of classification accuracy and strength for different biosignal types, which ultimately motivated its selection for FPGA deployment.

3.5 Data Collection Procedures

This section emphasizes the processes that followed for data acquisition and prepricing to support the training and evaluation of the 1-D CNN model. The dataset used includes ECG, EEG and EMG signals, each with separate sampling rate and feature dimensions. To ensure stability in the input, all signals were shaped and transformed to certain length, redesigned to adjust the sampling frequencies, and formatted to match the input requirements of the CNN model designed for the time series classification.

The full dataset was divided into three subsets: 70% for training, 15% for validation, and 15% for testing. The training set was used to iteratively update the CNN model parameters, while the validation set helped tune hyperparameters and monitor performance to avoid overfitting. The test set was reserved for evaluating the final model's generalization on unseen data.

Normalization was used to standardize feature values in the dataset, either by scaling to range of [0, 1] or standardization for zero mean and variance. During the training, this prepreparing step improved learning efficiency and model convergence.

Through these steps, an integrated and clean data set was designed to train and validate the 1-D CNN model in multiple biosignal types. The following sections are broad on design, training processes and hardware implementation.

3.6 Data Analysis Methods

This section provides an approach that was followed to analyze the data collected for the development and evaluation of the 1-D CNN model. This provides extension of the data involved in the study's analysis.

3.6.1 Model Training

Training the CNN model involved optimizing its weights and biases using the training dataset. Initially, the convolutional layers' parameters were randomly initialized. Input data was then propagated through the network via forward propagation to generate output predictions. The loss between predicted and actual labels was calculated using the cross-entropy loss function.

During backpropagation, gradients of the loss were computed with respect to the model's parameters, which were updated using Stochastic Gradient Descent (SGD). Various learning rates and momentum values were tested to identify the most effective training configuration. To improve generalization and reduce overfitting, regularization techniques such as dropout and weight decay were used.

3.6.2 Model Evaluation

The performance of the CNN model was evaluated using validation and testing datasets. Several metrics were calculated to assess the model's effectiveness:

1. Accuracy: The proportion of correctly classified samples out of the total samples was calculated. This provides a general measure of the model's performance.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.8}$$

where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives. The accuracy attained by each algorithm based on the neural network (NN) of the proposed model is shown in Figure 3.6.

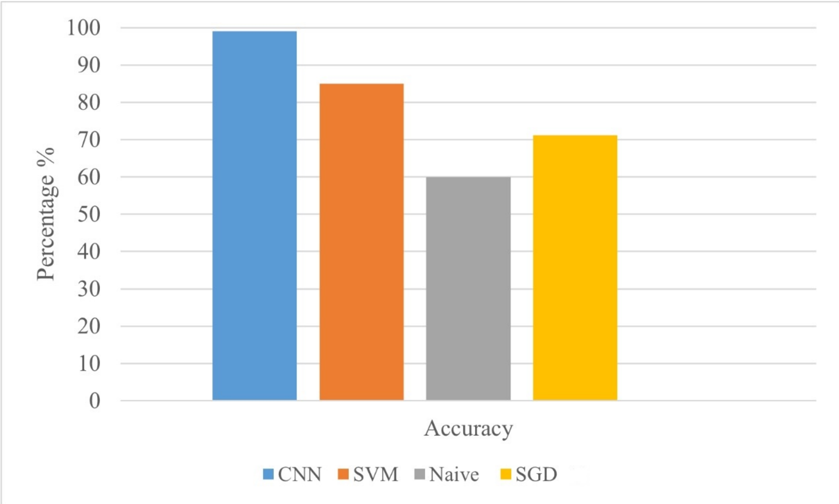


Figure 3.6. Accuracy of the 1-D CNN model when using a threshold probability for positive classification.

2. Precision: calculated by the number of true positive predictions divided by the total predicted positives. This shows how accurate the positive predictions are.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.9}$$

Figure 3.7 (c) describes the precision of the proposed model for each algorithm.

3. Recall: Number of correct positive predictions divided by the number of all actual positives. This is equivalent to the model being able to recall every relevant case.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.10}$$

Figure 3.7 (d) illustrates the recall of the proposed model for each algorithm.

4. F1-Score: The harmonic mean of precision and recall was computed to provide a balanced measure of the model's performance.

$$F1 = \frac{2}{\left(\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}\right)} \tag{3.11}$$

Figure 3.7 (a) presents the AUC of the proposed model for each algorithm. A value near 1 signifies that the classifier is performing effectively. In contrast, a value approaching zero indicates that the model is entirely incorrect and classifying in the opposite manner. A value around 0.5 suggests that the classifier is making random guesses. The relationship between the false positive rate (specificity) and the true positive rate (sensitivity) is illustrated in Figure 3.8.

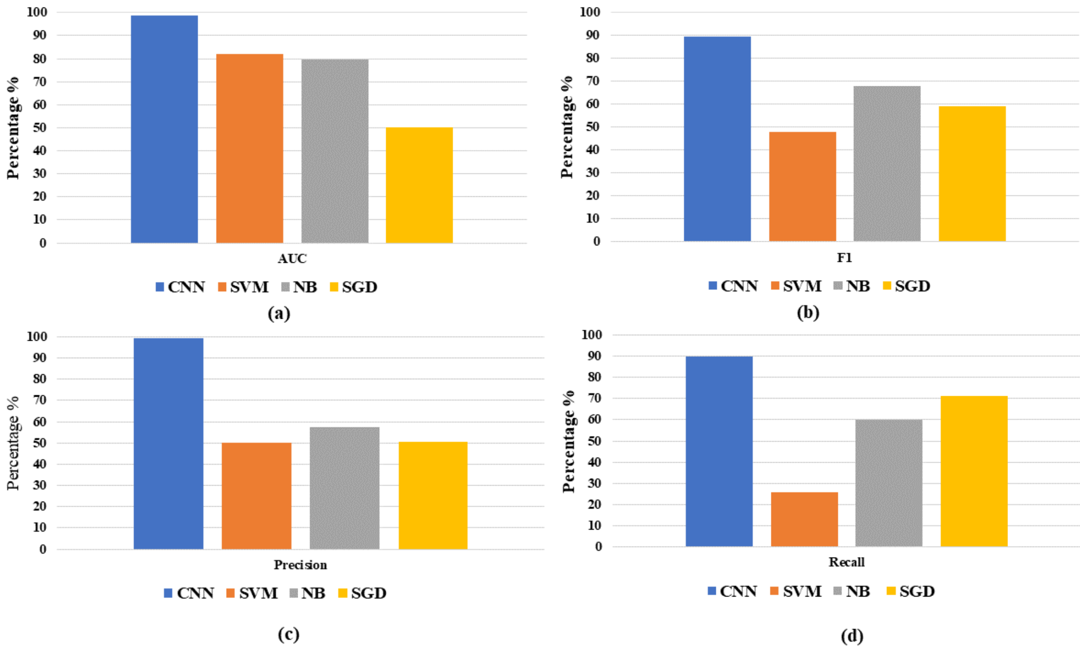


Figure 3.7. The performance of the first four models according to (a) Area Under the Curve, (b) F1-score, (c) Precision, and (d) Recall.

3.6.3 Cross-Validation

The model was trained on data and tested using cross-validation techniques to ensure it performs well and generalizes when predicting new or raw signals. The performance was consistent irrespective of how the data was split into training or test within each batch. K-fold cross-validation was employed, where the data was divided into five subsets, and trained and validated five times with a different subset as the validation set (15%) while training on the rest. This provided a more robust estimate of the model's performance.

3.6.4 Hardware Implementation Evaluation

The implementation of the 1-D CNN model on the FPGA platform was evaluated based on several key performance indicators:

1. Processing Speed: The speed at which the FPGA processed the data was measured and compared to other implementations. The proposed architecture achieved a high speed of 442.948 MHz, as shown in Figure 3.9 (a).

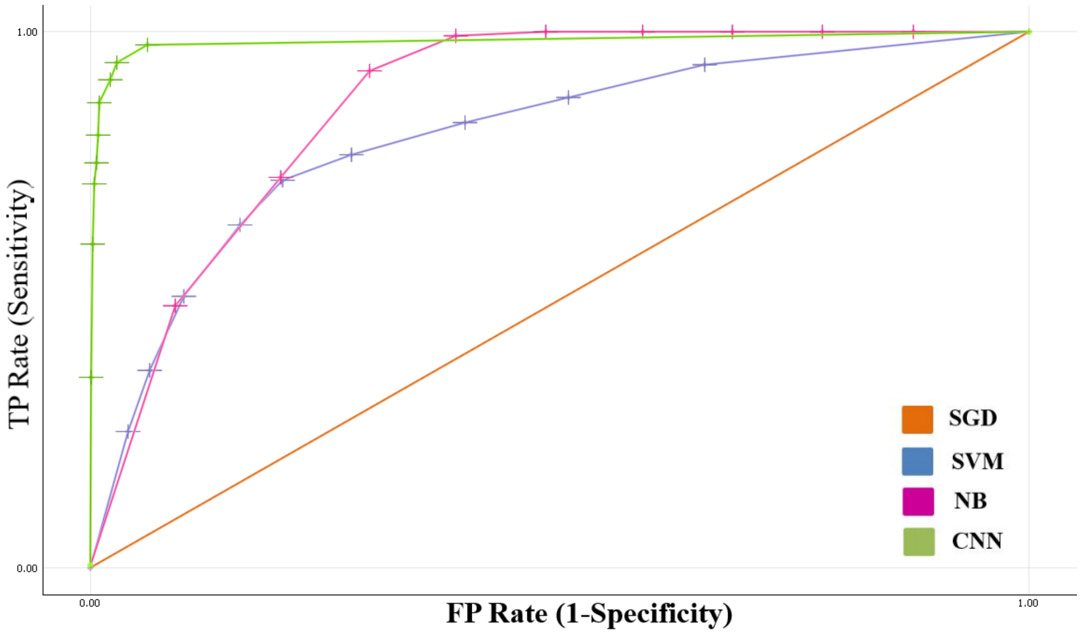


Figure 3.8. The ROC curves of the four models.

2. Resource Utilization: The amount of FPGA resources, such as logic units and memory, used by the model was recorded. The proposed architecture achieved resource utilisation of 1.068 KLUT, as depicted in Figure 3.9 (b).

3. Energy Efficiency: The power consumption of the FPGA during model execution was measured to assess energy efficiency. The proposed architecture demonstrated 161 GOP/s/W energy efficiency.

4. Throughput: The number of operations per second (e.g., Giga Operations per Second or GOPS) achieved by the FPGA implementation was calculated (as mentioned in Equation (3.7)). The proposed design achieved peak throughput of 1145 GOPS.

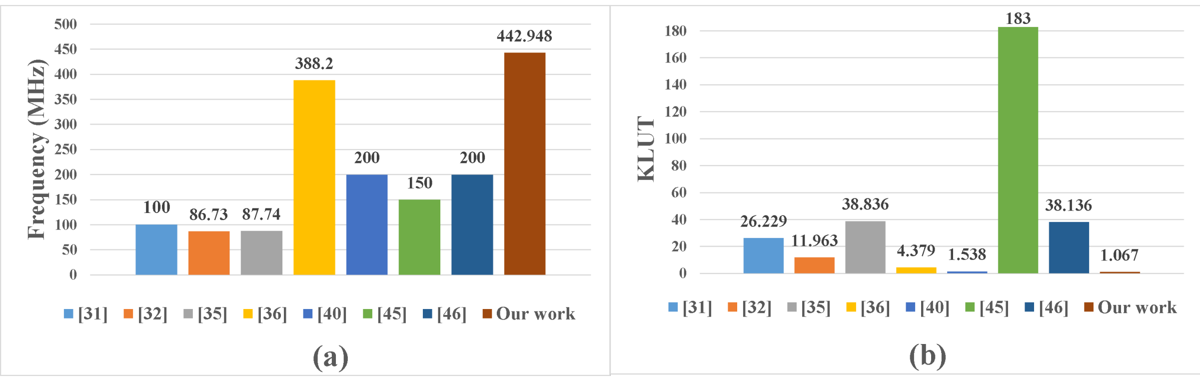


Figure 3.9. Illustration of the proposed 1-D CNN accelerator: (a) Comparison of operating frequency between the proposed structure and other models; (b) Comparison of resource utilisation (KLUT) among the models.

These results validate the effectiveness of the methodology introduced earlier in this chapter, where signal preprocessing ensured standardized inputs, shift-based PE design minimized hardware load, and the pipelined dataflow enabled real-time classification. The combined impact of these steps is reflected in the high operational frequency (442.9 MHz), low resource utilization (1.068 KLUT), and strong classification metrics (F1, AUC), demonstrating how each methodological design choice translated directly into performance gains.

To bridge the software and hardware stages, the trained CNN model parameters, including weights and biases were exported from Python as .txt files. These files were formatted to match the data interface of the FPGA system and uploaded to the off-chip memory prior to inference. At runtime, the PE arrays access these parameters from memory, enabling real-time classification without requiring on-chip learning. This separation of training and inference aligns with edge computing goals by ensuring low-latency and resource-efficient deployment on hardware.

3.7 Limitations of the Methodology

Several limitations were encountered during the development and evaluation of the 1-D CNN model implemented on the FPGA platform. One primary limitation was the hardware constraints of the Xilinx Zynq xc7z045 FPGA. Although this platform is efficient, its limited resources in terms of logic units, memory, and processing power restricted the complexity and depth of the CNN model that could be implemented. This variability was another challenge because the characteristics of each signal type was not always consistent between patients; the signals acquired with ECG, EEG, and EMG differed from one another patient to patient. These signals contain additional inter-subject and intra-subject variations that can influence the CNN model's performance.

Normalization and pre-processing of these signals to get standard results was a time taking and detail oriented job. Another limitation was concerned with the selection and quality of the development and test corpora. There are certain differences between the datasets available in public domain that can be observed in the signal quality, the sampling rate, and the free-from noise. These variations can bring in biases and distortions in generalization of the model, on cross datasets to ensure a good performance.

The applicability of the model on any other population also raised some concern. These datasets were collected on some certain population which can be a limitation in terms of generality. This limitation compromises the model's validity since it cannot generalize the results obtained to other demographics with different physiological traits. The real time computations were hindered substantially by the data movement rates between on-chip and off-chip memories. Although, the FPGA platform offers main enhancements in computing rate and power utilization.

Finally, in this study, it is clear that no all aspects of model performance have been described. Other measures like accuracy, precision, recall, F1-score, and AUC are equally informative, yet they may not reveal the whole picture of the model's effectiveness in more complex realities involving additional conditions. Thus, recognizing such restrictions, a clearer idea of what the researcher encountered during the investigation process will be presented. These findings are useful for future researches to investigate and enhance the performance and utility of CNN models in the classification of ECG, EEG, and EMG signals of wearable healthcare gadgets.

3.8 Conclusion and Summary

In this chapter, the proposed 1-D CNN architecture for the classification of ECG, EEG and EMG signal data obtained from the database and detection of various cardiovascular diseases is discussed. The key ones are the architectural design of the CNN model that would be highly suitable for FPGA implementation, the methods of biosignal data preprocessing and handling, as well as comprehensive assessment of the model's performance based on multiple factors. From the results, it can be concluded that by using the proposed 1-D CNN model, high classification accuracy could be obtained while consuming a small amount of energy and effectively managing the given resources. The proposed 1-D CNN accelerator works very efficiently due to using a tristate buffer in the multiplexer and replacing the multiplier by shift, which results in a resource-efficient accelerator.

The observed improvements in speed, energy efficiency, and accuracy were a direct outcome of the methodological innovations, namely the lightweight CNN architecture, shift-based arithmetic design, and pipeline scheduling in FPGA. These aligned perfectly with the system-level goals for wearable healthcare deployment, validating the proposed design both theoretically and practically.

Chapter 4

Implementation and Evaluation of Digital Twin Framework for IoT-Based Healthcare Systems

4.1 Introduction

Since the emergence of digital technologies there has been advancement at a very high rate such that different segments of the economy have benefited from it and this include the healthcare segment. The combination of the cloud environment, big data processing, and the IoT has created application opportunities primarily focused on advancing the healthcare system and enhancing patient wellbeing. One of such innovations is DT which entails the development of an information model that mirrors the physical entity for purposes of monitoring and even adjusting to real time conditions with the aim of preventing negative occurrences.

In healthcare, DTs present a new form of intervention by recreating the patient's physiological conditions in a virtual environment. This fosters constant tracking and evaluation of health aspects in the body hence early doctor interoperation and tailored care. However, it has been identified that the use of DTs in healthcare is still early stages. These are some current challenges of data integration, the problem of interoperability of systems, the issue of real-time processing. In this chapter the details of the infrastructure of the cloud-based DT ecosystem with the focus on real-time health monitoring and predictive analytics are outlined. Some of the factors which were measured by using the multiple sensors include the oxygen saturation (SpO2), heart rate (HR) and body temperature (BT); this system was connected to the cloud platform. This structure offers and supports the computation facilities for data storage, processing, and analysis. The primary contributions of this study are:

1. A new DT architecture is proposed, utilising cloud-based technology and healthcare

wearable devices, combined with a Pyomo-based dynamic optimisation model. This framework forms the basis of a Digital Twin Healthcare (DTH) application and effectively tackles challenges related to real-time monitoring, enhances system scalability, and improves resource management, while also boosting the precision 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 JSON-LD and sensors, for monitoring and health tracking in humans, utilising pay-as-you-go cloud services.

4. This study also seeks to verify the design based on the integration of physical and digital data using time series insight (TSI), with Flask used to validate the ML model. Latency calculation was also performed and in comparison with previous study, the obtained latency is low.

5. Establishment of a Robust Security Framework: By employing hybrid encryption, certificate-based authentication, Merkle Tree verification, and secure communication protocols, the system ensures enhanced data security and integrity throughout the data transmission process.

The remaining sections of this study are structured as follows: A cloud-based DT architecture is proposed in Section 4.2. Section 4.3 describes the framework developed on the basis of twin graph. The implementation setup is presented in Section 4.4. In Section 4.5 presented approach of Cybersecurity in digital twin healthcare step by step with explain algorithm of security. Section 4.6 describes Proof of Concept (PoC). The Results and Discussions is presented in Section 4.7. Finally, Section 4.8 provides the conclusion and summary for this study.

4.2 System Architecture

The architecture of DT ecosystem is designed across the cloud for the effective implementation of the entire health monitoring as well as the analysis for the predictive system. This architecture is multilayered, and every layer is responsible for certain aspects in the working of the system. All these layers collectively help in effective data acquisition, communication, analysis, and visualisation.

4.2.1 Proposed structure of Digital Twin Healthcare

A novel DT-based healthcare monitoring system utilising cloud computing was developed in this study. The system leverages DT through PaaS, providing digital monitoring as an advanced, computer-oriented solution. In Figure 4.1, the proposed architectural model for a DT, employing both edge computing and cloud computing, enables the analysis of mobile data while using the Azure cloud as the platform. The architectural framework is supported by the cloud platform and comprises five distinct layers.

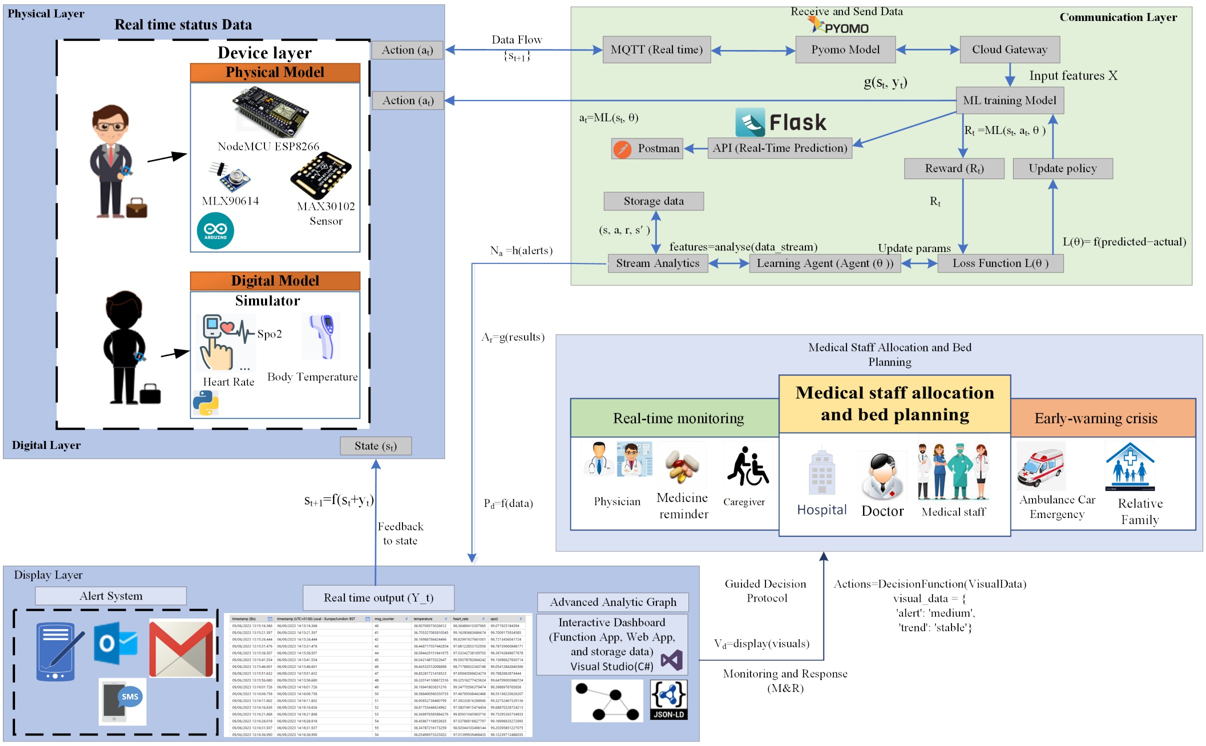


Figure 4.1. Proposed DT architecture including cloud, device, communication, and display layers.

The cloud platform supports the architectural framework, which consists of five distinct layers, enabling the practical integration of cloud and edge computing into the DT. This is demonstrated in Algorithm 4.1, which is designed to monitor and predict patient health indicators using a DT paradigm. The process begins by establishing the system through the integration of IoT devices within a specified starting period. The primary inputs include health metrics and historical health data. Data is processed to calculate the instantaneous fluctuations and disparities in health parameters. Once a specified data threshold is reached, the range and differences of the metrics are computed. The program then evaluates the health measures against predetermined thresholds.

A. Device Layer

On the left side of Figure 4.1, the device layer of the design architecture is presented. This layer includes the controller, a NodeMCU ESP8266, along with the sensors worn by an individual. The controller is essential for transmitting data to the IoT Hub using the Message Queuing Telemetry Transport (MQTT) protocol. MQTT, a messaging protocol specifically designed for networks with limited bandwidth, high latency, and inconsistent connectivity, is frequently encountered in IoT environments. Real-time communication is vital within the DT paradigm. Figure 4.1 illustrates HR, SpO2, and BT readings, which are sampled and sent to the communication layer. The sequential operating procedures from an implementation standpoint are depicted in this layer, which can be summarised as follows.

1. For the experiments, a NodeMCU ESP8266, a Max30102 sensor for SpO2 and HR, and an MLX90614 sensor for BT were utilized. The Max30102 sensor comprises internal LEDs, optical components, and photodetectors. It operates on a 1.8-V single power supply, with an additional 3.3-V supply for the internal LEDs.

2. Pulse oximetry involves the use of LEDs to interact with a photodiode placed on the patient's body.The calculation of SpO2 is expressed as follows:

$$SpO_2 = \frac{\mathcal{A}_{C_{\text{red}}} / \mathcal{D}_{C_{\text{red}}}}{\mathcal{A}_{C_{\text{IR}}} / \mathcal{D}_{C_{\text{IR}}}} \tag{4.1}$$

   This equation represents the ratio of absorbed to transmitted light at the specified wavelengths, thereby determining the oxygen saturation level in the blood.

3. Multiple measurements, including BT, HR, and SpO2, are recorded by the IoT node. The IoT node periodically records each parameter to ensure accuracy and reliability.

4. In this approach while IoT Hub does not conform to the formal MQTT broker protocol, it uses Device-to-Cloud (D2C) message transmission. In the following way it also enhances the system robustness and also provides assurance to the organizations that the cloud infrastructure is built strong. As one of the outstanding characteristics of the IoT Hub, there is a free messages option of up to 8000 daily, making this approach almost-free.

5. Data collection and transfer to the IoT hub are performed by the NodeMCU ESP8266. Once a connection with the cloud has been initiated and successfully created, the serial monitor will display an 'OK' message. Thus, this verification step not only confirms that all the procedures have been executed successfully but also increases the reliability of the system.

In summary, ESP8266 NodeMCU is an important part of data acquisition and data transfer in the Internet of Things environment. The printout of 'OK' on the serial monitor indicates that the cloud has been established and the correctness of the procedures for data transmission and the quality of the data that has been gathered are correct.

Algorithm 4.1. Intelligent Patient Surveillance and Predictive Analytics with DT

procedure MAIN
  Initialization: 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**
      $\mathfrak{I}_{df} = \mathcal{D}_r \times \frac{(\mathcal{M}(c_0)-1)}{\mathfrak{T}_s}$ ▷ 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 procedure

B. Communication Layer

The communication layer enables real-time data exchange between the device layer, edge layer, and cloud layer. Implemented protocols within this layer include HTTPS and MQTT, as illustrated in Figure 4.5. The communication layer is composed of three parts, which can be expressed as follows:

$$C_{\mathcal{L}} = (C_{\mathcal{L}-\mathrm{PD}}, C_{\mathcal{L}-\mathrm{VD}}, C_{\mathcal{L}-\mathrm{PV}}) \tag{4.2}$$

In this expression, $C_{\mathcal{L}-\mathrm{PD}}$ represents the communication between physical objects and DT data ($\mathfrak{D}_{DT}$), $C_{\mathcal{L}-\mathrm{VD}}$ denotes the interaction between virtual objects and $\mathfrak{D}_{DT}$, and $C_{\mathcal{L}-\mathrm{PV}}$ signifies the communication between physical and virtual objects. The $\mathfrak{D}_{DT}$ model comprises data from both physical and virtual objects. During communication, the data source, value, unit, and sample size are critical elements that influence the data being transmitted and received.

$$\mathrm{s_{t+1}} = \mathrm{f(s_t, a_t, e_t)}, \tag{4.3}$$

where $\mathrm{s_{t+1}}$ indicates the next state, $\mathrm{s_t}$ represents the current state, $\mathrm{a_t}$ is the action taken according to the policy, and $\mathrm{e_t}$ denotes an external environmental factor. This system evolves over time, influenced by actions and external factors. Machine learning models select actions to maximize perceived rewards:

$$a_t = ML(s_t, \theta), \tag{4.4}$$

where $\mathrm{a_t}$ is the action chosen by the machine learning model based on the current state $\mathrm{s_t}$ and model parameters θ. The rewards are calculated as follows:

$$R_t = ML(s_t, a_t, \theta). \tag{4.5}$$

These rewards R_t, based on the current state s_t, action a_t, and parameters, continually refine the model's policies using the update equation:

$$\theta_{\mathrm{new}} = \theta_{\mathrm{old}} + \alpha \nabla_\theta J(\theta), \tag{4.6}$$

where θ_{new} and θ_{old} are the updated and previous parameters of the model, respectively, α is the learning rate, and $J(\theta)$ represents the objective function, typically the expected reward. The system's performance is evaluated with a loss function:

$$L(\theta) = (R_t - \hat{R}_t(\theta))^2. \tag{4.7}$$

Where $L(\theta)$ is the loss function, R_t is the actual reward received, and $\hat{R}_t(\theta)$ is the predicted reward based on the model parameters θ. The learning agent performs stream analytics (feature extraction):

$$X_t = \mathrm{analyse}(s_t) \tag{4.8}$$

Where X_t represents the features extracted from the state s_t by the stream analytics process. This guides further refinements to model parameters through the learning process:

$$\theta_{\mathrm{new}} = \theta_{\mathrm{old}} - \eta \frac{\partial L}{\partial \theta} \tag{4.9}$$

Where 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. This process balances exploration and exploitation in decision-making. To sum up, the communication layer's protocols and structures

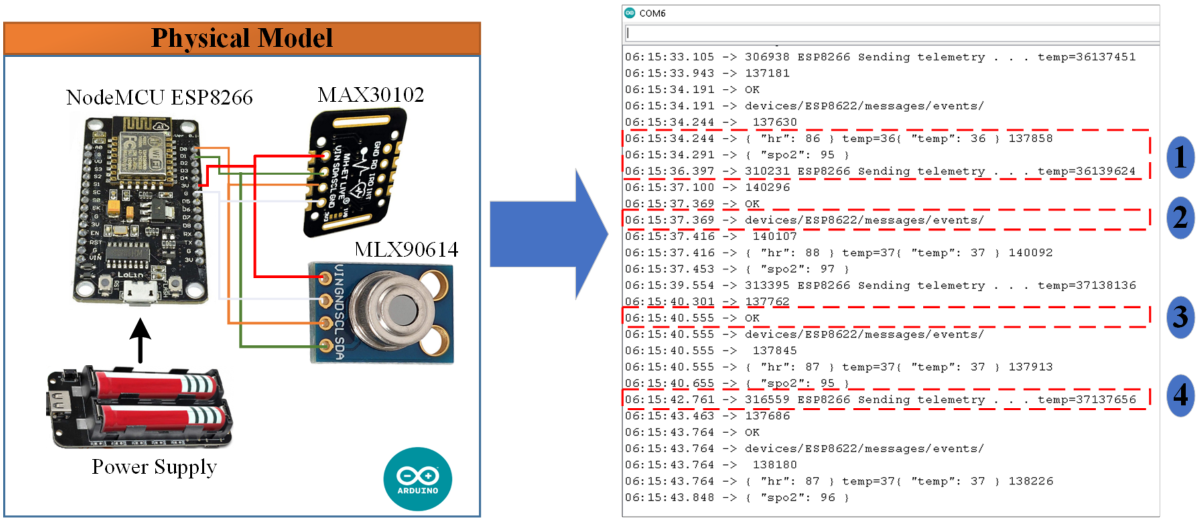


Figure 4.2. Sensor output in the IoT device framework.

enable seamless real-time data exchange, critical for the efficient operation of the digital twin ecosystem in healthcare.

C. Digital Model Layer

The digital model layer is designed to provide users with medication reminders and emergency alerts while continuously monitoring a patient's physiological status through wearable devices. This layer involves constructing a participant's historical data, sensor models, and behavioural models to characterize activities such as medication adherence and emergency responses. These models can predict future actions and facilitate evaluation, reasoning, and prediction using rules of association, constraints, and deductions, as illustrated in Figure 4.1. The Digital Twin for Healthcare (DTH) model is expressed as:

$$DTH = (\mathcal{P}_i, \mathcal{S}_m, \mathcal{B}_m, \mathcal{F}_m, \mathcal{N}_c) \tag{4.10}$$

where $\mathcal{P}_i$ includes personal information such as name, gender, age, and historical data. This information forms the basis for managing personal health. The component $\mathcal{S}_m$ primarily consists of medical sensor data, including HR, SpO2, and BT.

A behavioural model, $\mathcal{B}_m$, is designed to characterize the state of an individual, whether a patient or an elderly person, to monitor their status, including the amount of medication taken or crisis behaviours such as cardiac arrest or respiratory arrest. The element $\mathcal{N}_c$ describes the network connection between the cloud environment and the devices.

Model evolution occurs in parallel, with models being calibrated to run synchronously with physical objects. The advanced models provide more precise estimation, optimisation, and forecasting of the operational process model, denoted as $\mathcal{F}_m$.

D. Cloud Layer

The cloud layer furnishes the necessary computational resources for data storage, processing, and analysis, thereby supporting the extensive data handling needs of the DT ecosystem. High-performance computing resources within the cloud facilitate the execution of complex machine learning algorithms, which are crucial for predictive analytics. The cloud infrastructure ensures scalability, enabling the system to manage large data volumes from multiple sources without performance degradation. To connect time series insights and Azure Digital Twin (ADT) with the IoT hub, a bridge is required to transmit data from the cloud to the display layer for monitoring purposes. In the current study, Digital Twin Definition Language (DTDL) employs JSON-LD, an open language similar to JSON. The DTH platform enables remote monitoring and health tracking, scalable for both smart devices and patients. An IoT hub transmits patient monitoring data to the cloud via a twin-graph platform. Serverless Functions apps reduce the need for extensive code, infrastructure management, and associated costs. Within the broader research framework, Algorithm 4.2 is identified as essential for achieving data synchronization between IoT devices and the Digital Twins environment.

This foundational role is further elucidated through a detailed exposition that follows:

1. Upon the algorithm initiation, basic namespaces are invoked: Digital Twins SDK for .NET for creating digital twins, the cloud identity to authenticate the user, and utilities for parsing IoT devices data.

2. ADT_URL that describes the URL of a digital twin instance endpoint. This URL is very important because it points the algorithm to the correct DT service.

3. This prevents the algorithm from being started in situations where environment variables. If the ADT_URL is undefined then all parameters needs to be initiated otherwise error log created.

4. Credential objects are created using the default cloud credential method for seamless authentication within the suite of cloud services. Subsequently, a connection is established with DT using these credentials, forming a secure and robust link with the Digital Twin Client object.

5. The UpdateDigitalTwin function is then invoked using the device's unique identifier and a dictionary of updated data, ensuring the latest sensor information is accurately reflected on the DT. This guarantees that the DT correctly represents the current status of the physical device.

Moreover, the algorithm is designed to handle a large number of events, which minimising latency and provides frequent updates of digital twins, and enhance of performance and increasing the number of events. The system's scalability that is evident from the communication protocols outlined in Table 4.1 shows that the architecture of the system is well equipped to deal with the increased volume of traffic and the load. Integration with the system architecture is achieved in Algorithm 4.2 which allows for updates to the digital twins and representation of changes in the physical environment. This algorithm works together with IoT Hub for device management and digital twin for modeling and simulating a real world, which is very useful for monitoring, analysing, and responding of this system as demonstrated the Proof of Concept (PoC) mentioned in the Subsection 4.6.2

Algorithm 4.2. Digital Twin Data Update

Start
Import necessary libraries
Input: $\mathcal{E}$ (EventGridEvent), $\mathcal{L}$ (log)
Output: Status of the update
$\mathcal{A}_U \leftarrow$ `getenv`$('ADT\_URL')$
if httpClient is true **then**
    `curl_easy_setopt(httpClient)`
end if
if $\mathcal{A}_U$ = null **then**
    $\mathcal{L}$.`LogError`("Error: 'ADT_URL' is not set")
else
    $C \leftarrow$ `Credentials()` % *Default cloud credential*
    $\mathcal{DTC} \leftarrow (\mathcal{A}_U, C)$ % *Digital Twin Client*
    $\mathcal{L}$.`LogInformation`$("ConnectedtoADT")$ % *Log the connection status*
    if $\mathcal{E} \neq$ null **and** $\mathcal{E}$.Data $\neq$ null **then**
        $\mathcal{DM} \leftarrow$ `JObject.Parse`($\mathcal{E}$.Data) % *Device Message*
        $\mathcal{DI} \leftarrow \mathcal{DM}$.`GetValue`$('IoTHubId')$.`ToString()` % *Device ID*
        $\mathcal{DS} \leftarrow \mathcal{DM}$.`GetValue`$('DeviceData')$.`ToString()` % *Device Data*
        $\mathcal{L}$.`LogInformation`$("DeviceID : " + \mathcal{DI} + ", DeviceData : " + \mathcal{DS})$ % *Log device information*
        $\mathcal{UD} \leftarrow$ `new Dictionary<string, object>()` % *Update Twin Data*
        $\mathcal{UD}$.`Add`$("DeviceData", \mathcal{DS})$ % *Add device data to dictionary*
        $\mathcal{DTC}$.`UpdateDigitalTwin`$(\mathcal{DI}, \mathcal{UD})$ % *Send update to Digital Twin*
    end if
end if
End of function

E. Display Layer

As illustrated in Figures 4.1 and 4.2, the importance of the display layer within the DT system, primarily based on a cloud platform, is highlighted. This layer fundamentally integrates Time Series Insights (TSI) for data analytics and storage for data preservation. Conversely, the ADT Explorer is designed specifically for the visual exploration and management of DT. Function Apps play a crucial intermediary role, facilitating the connection between TSI and the Twin Graph. This connection enables efficient data flow and processing through Visual Studio's capabilities. At the core of the architectural framework is DTDL, which defines the models of the DT, while the Twin Graph visually depicts the interconnections between these models. The layer is optimized with a variety of output channels, including JSON-LD for data exchange, SMS for immediate alerts, and notifications via mobile, online, and email interfaces.

Real-time data from various inputs, such as HR, SpO2, and BT monitors, is aggregated at both physical device and DT layers. This data is encapsulated within a state $\mathrm{s_t}$ and transitions based on internal dynamics and external feedback, as shown by the equation:

$$s_{t+1} = f(s_t, y_t) \tag{4.11}$$

which showcases the system's adaptive capabilities. The data is processed:

$$P_d = \text{process(data)} \tag{4.12}$$

data analysis is denoted by:

$$A_r = \text{analyse(data)} \tag{4.13}$$

and monitoring for critical thresholds to generate alerts is given by:

$$N_a = \text{alert(data)} \tag{4.14}$$

with all outputs displayed in a comprehensive alert system designed to assist medical staff in real-time decision-making:

$$V_d = \text{display(visuals)} \tag{4.15}$$

This integration not only supports immediate responses but also creates a continuous feedback loop, thereby improving the system's accuracy and responsiveness. In addition, healthcare services involving DT data encompass the physical and digital states of items, as well as information on services and the fusion of these two states, as illustrated below:

$$\mathfrak{D}_{DT} = (\mathcal{D}_{PA}, \mathcal{D}_{DO}, \mathcal{D}_{HS}, \mathcal{D}_{F}) \tag{4.16}$$

where $\mathfrak{D}_{DT}$ represents DT data from both 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 historical data from hospital records at time t_0, and $\mathcal{D}_F$ is the fused data. The equivalent value at the current time, denoted by, is expressed as:

$$\Delta\mathfrak{D}_{DT}+ = \mathfrak{D}(t_0) - \mathfrak{D}(t_0 - 1) \tag{4.17}$$

The responsibilities for data processing, which illustrate data transfer and processing in the proposed architecture, are explained in Subsubsection 4.2.1.

4.3 Framework Established Based on Twin Graph

In Figure 4.3, a patient is connected to two sensors via a knowledge graph, allowing for the integration of multiple sensors and monitoring capabilities. An integrated environment is created for monitoring SpO2, HR, and BT in the cloud. Machine learning is employed to predict patients' future states, showcasing the development of a novel architecture with multiple applications. Figure 4.4 depicts the implementation through the use of ADT Explorer.

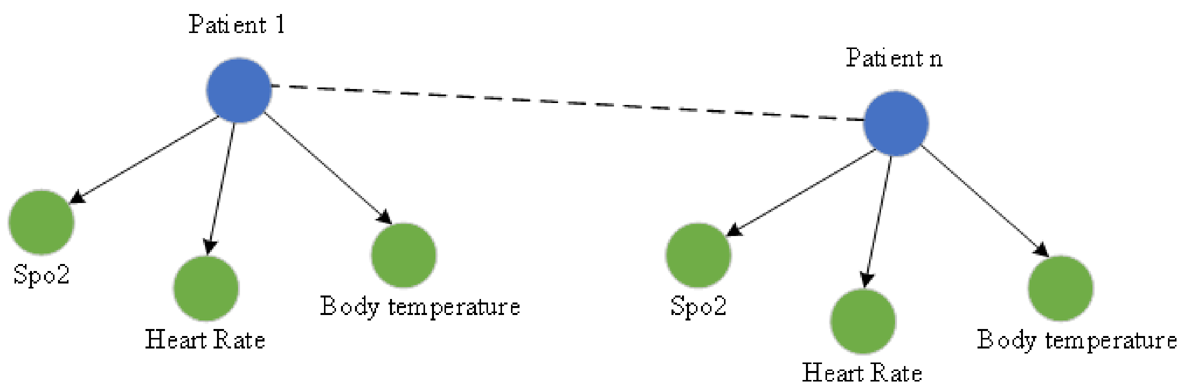


Figure 4.3. Configuration of DT utilizing twin graph.

Additionally, the provided code establishes a digital interface that enables the monitoring of BT, SpO2, and HR, which is a crucial component of the system. In Section 4.4, machine learning is used as an example to demonstrate how to develop a novel architecture with several applications. The process begins by creating a DT instance, implementing role-based authentication, and feeding the DT with data from IoT and edge devices via REST APIs. Once a DT instance is established, the environment's operator can assign specific roles to members and doctors. ADT facilitates queries, model modifications, and data visualization using Explorer. ADT models are authored in DTDL and stored as JSON-LD files.

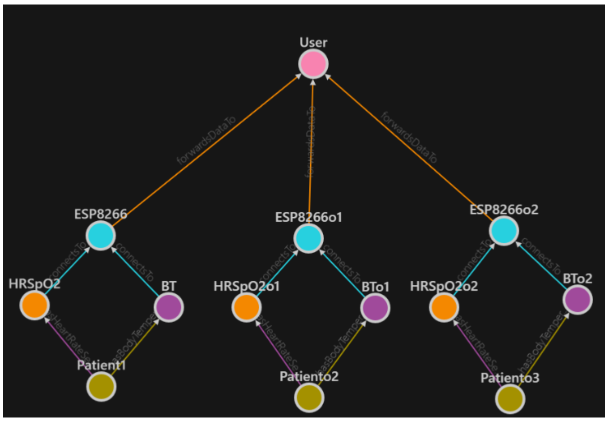


Figure 4.4. Deployment of DT in ADT explorer.

4.4 Implementation Setup

In this study, a DT platform for intelligent healthcare systems was developed, utilising the cloud, wearable devices, data analytics, and machine learning to create virtual patient replicas. This approach enhances collaboration and enables remote patient monitoring. The platform consists of both a physical and a virtual element, each elaborated upon in detail in their respective sections. The functionality and implementation of the physical element were described according to Algorithm 4.1. The virtual element's establishment, including data transfer, storage, processing, and decision-making processes, were outlined in Algorithm 4.2. Additionally, the procedure for managing the implementation process will be demonstrated. Machine learning techniques were employed to train the model using seven different algorithms, from which the optimal predictive algorithm was selected and classified on the ML platform.

In Figure 4.5, the sequence diagram illustrates the communication exchanges and interactions among various system entities. This diagram effectively portrays the generation, validation, and processing of telemetry data by different components, providing a comprehensive view of establishing a DT environment for healthcare. It outlines the mechanisms for managing bidirectional data transmission. Telemetry data is received by the user's device from a client source, validated by the "Data Validation" participant, and then transmitted to multiple recipients. The diagram includes a "Main loop" iteration, which handles errors via the "Error Handling" participant and displays data through the "Real Time Visualisation" participant. The deployment of the ML model is performed by the "ML Model" participant, while data archiving is handled by the

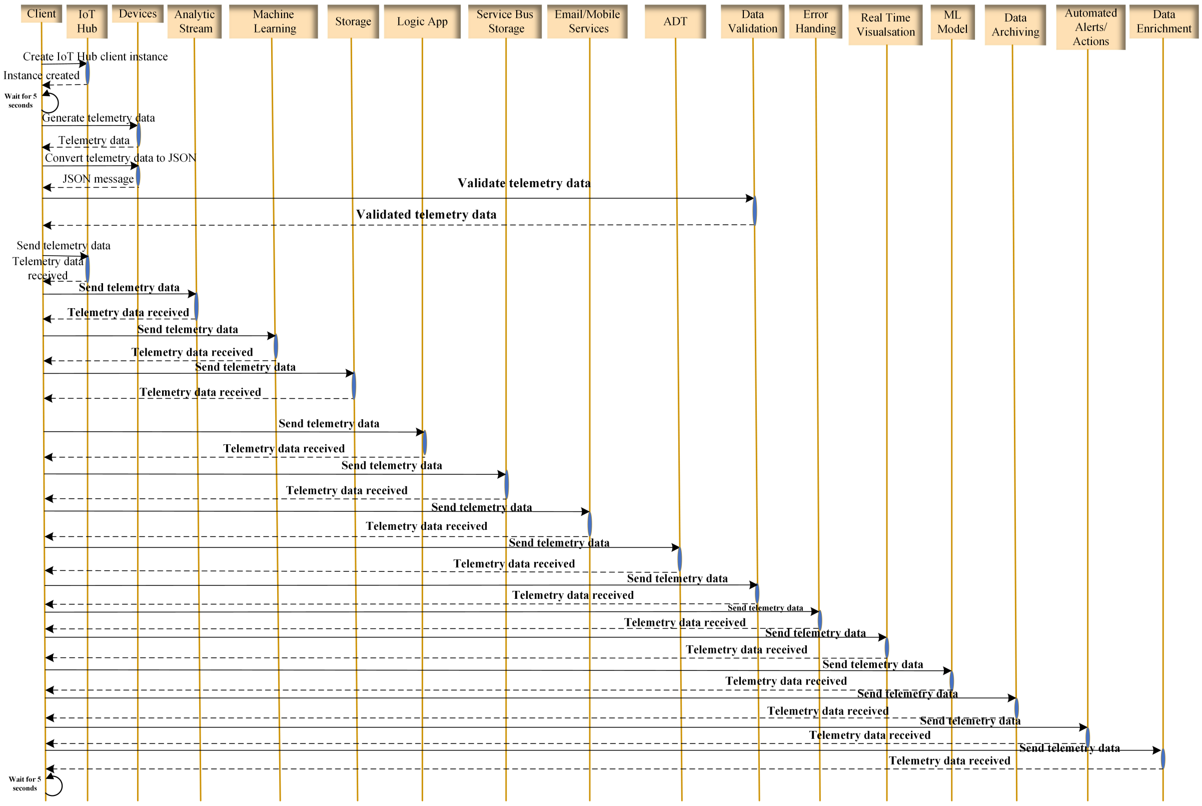


Figure 4.5. Diagrammatic representation of the DT sequence model.

”Data Archiving” participant. Automated alerts and actions are triggered by the ”Automated Alerts and Actions” participant. The telemetry data is enhanced by the end user through the ”Data Enrichment” participant, resulting in an ”Enriched telemetry data” message.

Table 4.1. Network Configuration Parameters

Parameter	Value
Scenario	Indoor/Outdoor
Channel Band / Bandwidth	2.4GHz / 20MHz
BS/UE Tx Power	20/15 dBm
Traffic	MQTT, Sensor Data
Packet Size	1024 bytes
Data Transmission Frequency	2-8 Hz
Solver	GLPK (Linear Programming Solver)
Communication Protocols	MQTT over SSL/TLS, I2C
Security	TLS with X.509 Certificates

The specifications outlined in Table 4.1 illustrate the robust framework of our digital twin system, highlighting the emphasis on security through TLS with X.509 certificates and the use of efficient data transmission protocols like MQTT over SSL/TLS. Adjustable data transmission frequencies (2-8 Hz) allowed for flexibility in patient monitoring. The GLPK solver optimized telemetry frequency, improving system responsiveness, while Merkle Tree validation ensured data integrity.

A. 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 4.1, Pyomo's linear programming functionality dynamically adjusts telemetry frequency and prioritisation in response to network conditions, reducing latency and maintaining efficient CPU and network performance.

To ensure efficient use of computational and communication resources while maintaining low-latency data transmission, Pyomo based optimisation model was incorporated into the proposed DTH system. Pyomo, a Python based open-source optimisation modelling framework, was selected for its ability to handle both linear and mixed integer programming problems in real time.

In this context, Pyomo dynamically adjusts the telemetry transmission frequency for each physiological signal (e.g., HR, SpO_2, BT) based on system load, network constraints, and the urgency level of each metric. The model continuously monitors incoming data and assigns priority levels using real-time, rule-based conditions such as:

- If HR exceeds the critical threshold (e.g., >120 bpm) while SpO_2 and BT remain within normal ranges, HR is flagged as critical, and its telemetry frequency is increased.
- If all metrics remain stable, telemetry frequency is reduced to conserve bandwidth and processing power.
- If multiple metrics are abnormal, their transmission is optimised jointly based on severity.

Mathematically, this process is formulated as an optimisation problem. Let:

1. T_i be the telemetry frequency for physiological signal $i \in \{\mathrm{HR}, \mathrm{SpO_2}, \mathrm{BT}\}$,
2. $P_i \in \{0, 1\}$ indicate whether signal i is prioritised (1) or suppressed (0),
3. C be the total available communication bandwidth (e.g., bytes/second),
4. B_i be the bandwidth requirement of each signal per transmission.

The objective is to maximise the clinical relevance of transmitted data while minimising bandwidth usage:

$$\max \sum_i P_i \cdot \text{Urgency}_i - \lambda \sum_i T_i \cdot B_i \tag{4.18}$$

Subject to:

$$\sum_i T_i \cdot B_i \leq C \tag{4.19}$$

$$T_i^{\text{min}} \leq T_i \leq T_i^{\text{max}} \quad \forall i \tag{4.20}$$

$$P_i = \begin{cases} 1 & \text{if Metric}_i \notin \text{Normal Range} \\ 0 & \text{otherwise} \end{cases} \tag{4.21}$$

Here, Urgency_i is a dynamically computed weight reflecting the clinical criticality of metric i, and λ is a regularisation parameter that balances transmission urgency with bandwidth usage. For example, if HR is elevated but other metrics remain within normal limits, the model automatically suppresses transmission of less urgent data to preserve resources.

The integration of Pyomo enables the system to intelligently drop non-critical data, avoid congestion, and maintain reliable performance under varying network conditions. The benefits of this optimisation are evident in Figure 4.19 and Figure 4.20, which demonstrate significant reductions in latency and improved system efficiency. Algorithm 4.3 describes the process of telemetry optimisation and resource monitoring.

Algorithm 4.3. Telemetry Optimization and Resource Monitoring with Pyomo

```
procedure TELEMETRY TRANSMISSION AND OPTIMIZATION
    Initialization:
    Connect devices to IoT Hub
    Initialize Pyomo model for optimization (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 Optimization
    Define variables for telemetry frequency and priority in the Pyomo model
    Set constraints on frequency and priority ranges
    Solve the Pyomo model to minimize 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, optimizing transmission, and monitoring resources in the next cycle
end procedure
```

B. Data Acquisition

Accurate predictions were achieved through the utilization of diverse data, feature extraction, and historical information. Data were sourced from the MIMIC-III clinical database [161]. To validate the model, an application was developed for a group of volunteers, implementing the proposed DTH design as outlined in Section 4.2. Data were directly collected from these volunteers. Vital metrics were labelled using medically accepted ranges, with abnormal signs classified when they exceeded typical physiological limits. These metrics were obtained from academic literature and various reputable websites. Specifically, BT was considered normal within the range of 36.5 °C to 37.3 °C; heart rate was considered normal between 60 and 100 beats per minute; and SpO2 levels were deemed normal at 95% or higher [162], [163]. This rigorous labelling process ensured the reliability of the data used for training and validating the machine learning model.

C. Data Analysis and Pre-processing

The data used for modelling underwent thorough statistical analysis, including the creation of histograms and outlier graphs. Real-time data transmission from sensors and the NodeMCU may result in data loss or the presence of outliers. To address missing data, mean values or other statistical imputation methods were employed, and feature selection was used to reduce the input data. The Pearson correlation coefficient was utilized to examine variable dependence, as outlined in Algorithm 4.4. This algorithm processes three primary inputs: BT, HR, and SpO2. These inputs, derived from sensor data, are susceptible to outliers and missing values due to transmission losses or sensor inaccuracies. To ensure the accuracy and applicability of the data, several preparatory actions were undertaken. Normalization was used to regulate the volume of the datasets for the allowable analysis range while formatting checked whether the data structure complied with the analytic platform. The relationships between the variables were further determined by finding the coefficient of correlation by the Pearson method. Recognising dependencies between the features allows for the identification of which variables are not needed and give insight into the nature of the dataset.

D. Data Standardisation and Model Evaluation

To normalise the data the StandardScaler technology was used so that all features are proportional to one another. For the modelling process, 75% of data was used for training purpose and the 25% of data was used for testing purpose. This split was selected to give an objective assessment of the model's performance.

In this study, seven classical machine learning classification algorithms were used to assess and compare the results in this study [164], [165]. These algorithms are Random forest, Gaussian Naïve Bayes(GNB), logistic regression, K-Nearest Neighbour(KNN), Decision tree, XGBoost, Support vector machine(SVM). Every algorithm was chosen with respect to its characteristics and applicability to various aspects of the classification tasks. XGBoost is a highly efficient gradient boosting technique that is widely used for working with big data sets. The model was optimized 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.

Algorithm 4.4. Data preprocessing functional element

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

4.5 Approach of Cybersecurity in Digital Twin Healthcare

Figure 4.6 presents a comprehensive framework designed to fortify the cybersecurity infrastructure of a DTH system. At the core lies the safeguarding of patient data, around which the entire model is constructed. This is epitomised by the Patient Data Security segment, which encapsulates pivotal cybersecurity measures such as hybrid data encryption and certificate-based authentication. These measures are essential for ensuring the confidentiality and integrity of patient information.

Furthermore, the Merkle Tree integrity check and secure communication components provide a robust foundation for verifying the consistency and security of the data, while cyber threat prediction focuses on proactive identification of potential security breaches. Collectively, these elements form the bulwark against cyber threats, pivotal in maintaining the trust and safety required in healthcare environments.

The framework also delves into the organisation requirements specification, dissecting both the functional and non-functional requirements necessary for a thorough investigation into the system's necessities. Subsequently, cost model analysis evaluates the financial aspects, crucial for sustainable implementation.

In the assessment of cloud supporting step, the system's elasticity, communication, processing, and infrastructure control are scrutinised, along with availability and security protocols.

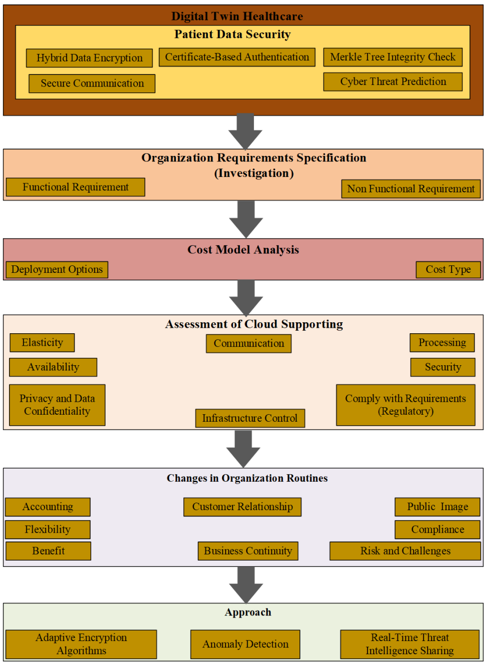


Figure 4.6. Integrated framework for digital twin healthcare cybersecurity.

This assessment is vital for ensuring compliance with regulatory requirements and upholding privacy and data confidentiality. To bolster security, a novel hybrid encryption technique has been developed, combining the strengths of symmetric and asymmetric encryption systems. This approach leverages the resilience of public-key encryption with the efficiency of symmetric encryption, enhancing overall security. The algorithmic procedure is outlined in Algorithm 4.5. Also, Algorithm 4.5 details the certificate-based authentication method, crucial for establishing trust in the heterogeneous IoT landscape. To ensure data integrity throughout its lifecycle, the research incorporates a Merkle Tree-based integrity validation technique. In addition, the handshake process ensures the efficiency and safety of data transmission by establishing the channels of communication. Moreover, the real-time data analysis capability inherent in DT allows for the prediction and prevention of security breaches. This ensures the safety and preservation of healthcare data, as detailed in Algorithm 4.5. In Figure 4.7, the cybersecurity process flow for DTH systems is illustrated. Initially, data is encrypted using a hybrid encryption technique that combines symmetric and asymmetric methods. This ensures that data is protected from unauthorised access.

Modifications in organisational routines are addressed in the subsequent module, reflecting the changes brought about by the integration of the DTH system. It encompasses alterations in accounting practices, customer relationships, public image, flexibility, business continuity, compliance, benefits, and the identification of potential risks and challenges.

Lastly, the approach step posits innovative approaches such as anomaly detection, enhancing the security framework. The integration of real-time threat intelligence sharing, and adaptive

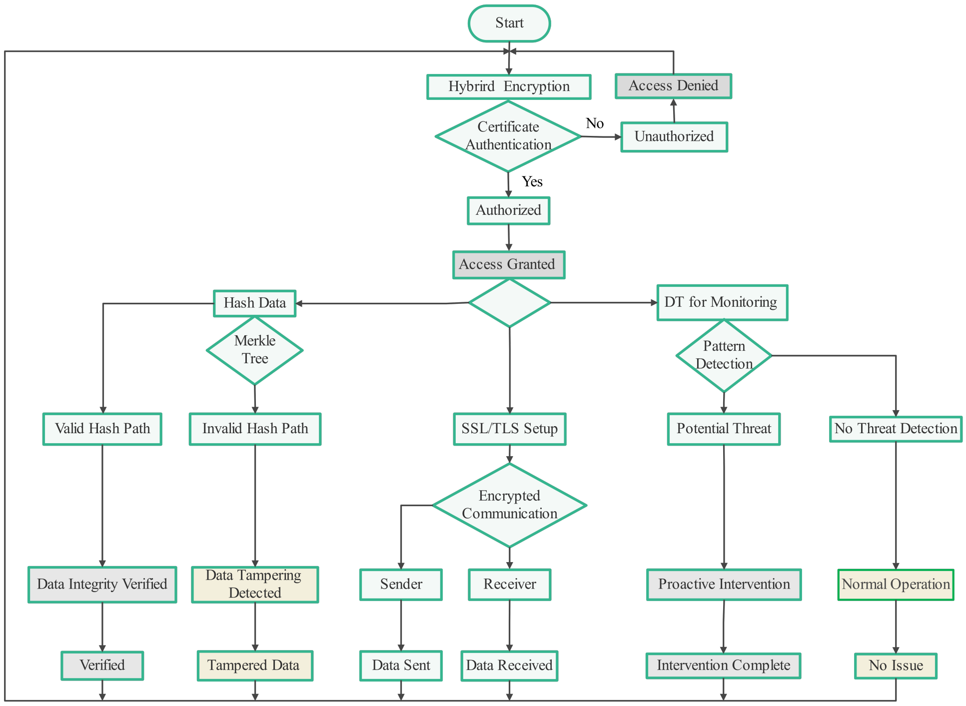


Figure 4.7. Cybersecurity process flowchart for DTH system.

encryption algorithms, for privacy preservation further contribute to the cutting-edge nature of the system.

In summary, the illustrated framework provides a strategic approach to securing DTH systems, ensuring comprehensive protection and resilience against cyber threats, while fostering advancements in healthcare technology.

Algorithm 4.5. Hybrid Security Framework for Encrypted Data Transmission with Integrity and Threat Detection

```
Input: Data D, Data Blocks DB[1...n], Public Key PK, Symmetric Key SK, Device Certificate DC, Certificate Authority CA, Secure Channel SC, Real-time Data RTD, Historical Data Patterns HDP, Merkle Root MR
Output: Transmission Status, Alert (if any)
function UnifiedSecureTransmission(D, DB, PK, SK, DC, CA, SC, RTD, HDP, MR)
    # Step 1: Authenticate the device
    status ← AuthenticateDevice(DC, CA)
    if status == "Failure" then
        return "Authentication Failed"
    end if
    # Step 2: Check data integrity using Merkle tree
    integrity_status ← CheckIntegrity(DB, MR)
    if integrity_status == "Invalid" then
        return "Data Integrity Check Failed"
    end if
    # Step 3: Encrypt the data using hybrid encryption
    ED, ESK ← EncryptData(D, PK, SK)
    # Step 4: Establish a secure communication channel
    TransmitStatus ← TransmitData(ED, SC)
    if TransmitStatus == "Not Sent" then
        return "Transmission Failed"
    end if
    # Step 5: Monitor real-time data for threats
    twin_data ← MIRROR_DATA(RTD)
    anomaly ← COMPARE(twin_data, HDP)
    if anomaly detected then
        GENERATE_ALERT
    end if
    return "Transmission Successful"
end function
```

4.6 Proof of Concept

4.6.1 Integration of IoT Devices in Healthcare Monitoring

Figure 4.2 illustrated the integration of IoT devices within a physical model, demonstrating the transmission of health data. The serial monitor on the left captures output from an ESP8266 module, including telemetry such as heart rate and temperature readings. These data points, which are timestamped sequentially, are transmitted to an IoT node, as indicated by the dashed red lines. The physical model on the right delineates the configuration of connected devices: the NodeMCU ESP8266 microcontroller, the MAX30102 pulse oximeter, and the MLX90614 infrared thermometer, culminating in the representation of a patient model. This setup exemplifies the practical application of IoT in patient health monitoring, showcasing the cohesive operation of sensor devices and data relay mechanisms as discussed in Subsection 4.2.1.

4.6.2 Schematic and Data Flow in Digital Twin Health Monitoring

Figure 4.8 demonstrated the schematic design of a health monitoring system within the digital twins environment. In this study, the Azure cloud platform was employed to implement the system. The ESP8266 Sensor Hub is shown as a central node, connected to the sensor interface and the saturation interface, indicating the collection of data from various sensor sources. The patient information interface, which serves as a hub for patient-specific data, is bidirectionally connected to both sensor interfaces, allowing for the reciprocal flow of health-related metrics. This model exemplifies the modular approach adopted in digital health solutions, where patient data is centralized, and sensor information is seamlessly integrated for comprehensive health management and monitoring.

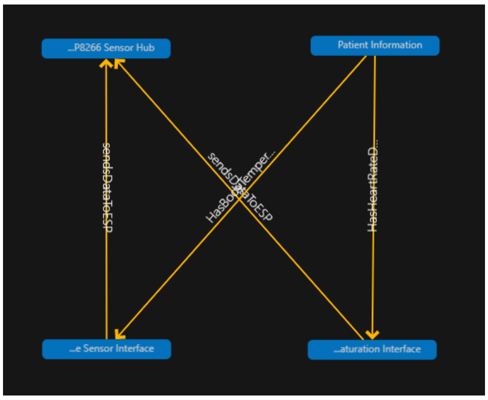


Figure 4.8. Configuration of the DT Model for health monitoring.

The interface within the digital twins explorer, depicted in Figure 4.9, shows sensor data from various health-monitoring devices. Blood oxygen saturation and heart rate, measured by a MAX30102 sensor, and body temperature, gauged by an MLX90614 sensor, are displayed. These metrics are updated dynamically, demonstrating the system's capability to monitor patient vitals. The immediacy of data acquisition is confirmed by the timestamps associated with each measurement, indicating the potential for this system's deployment in continuous health monitoring applications.

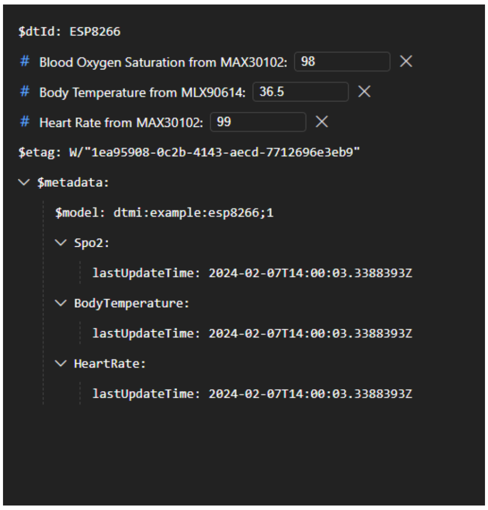


Figure 4.9. Live data feed from health monitoring sensors.

Figure 4.10 shows the invocation logs of a function app, which acts as an intermediary between the IoT Hub and DT. From the logs, it can be observed that upon receiving an event trigger, a connection to DT service is successfully established by the function. The logs provide details about the device ID, the authentication method, and the precise time the message was enqueued, confirming the processing of telemetry data. The encoded message body, which contains vital sensor readings, indicates that the data is subsequently decoded and applied within DT environment to update the twin's state.

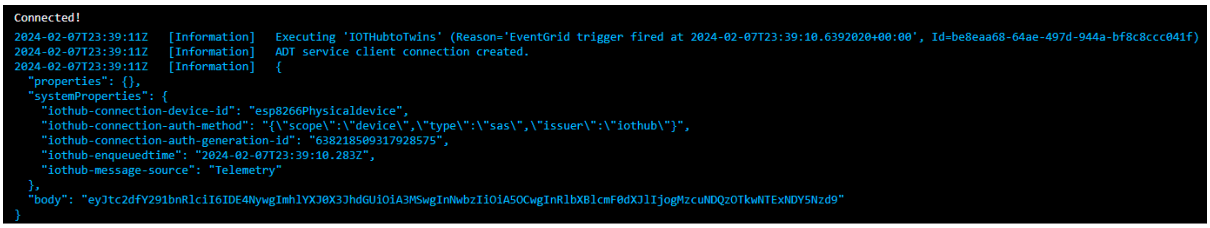


Figure 4.10. Invocation logs for function app synchronizing IoT and DT.

4.6.3 Analysis and Visualisation of Biometric Data in Digital Twins

Figure 4.11 illustrates a tabulated excerpt of biometric data extracted from TSI. The columns display timestamped readings of BT, HR, and SpO2, each associated with corresponding event counts. The data, organised chronologically, reflect the continuous telemetry captured from patient monitoring sensors. This table supports the data structure used for subsequent temporal analysis.

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 4.11. Tabulated biometric data from time series insights.

In Figure 4.18, a graphical representation of biometric readings over time, visualized in TSI. Trends for heart rate, SpO2, and body temperature are plotted against time, illustrating the fluctuations and stability within the captured data streams. Each line represents a different biometric parameter, providing a comprehensive view of the patient's physiological status. This visualization is crucial for validating the monitoring capabilities of the deployed digital health solution, as discussed in Subsection 4.2.1.

4.7 Results and Discussions

4.7.1 Model Evaluation and Comparison

This study compared seven ML models to determine the most suitable algorithm for real-time deployment in Digital Twin Healthcare (DTH) systems. The evaluation considered accuracy, precision, recall, F1 score, AUC, computational time, and cross-validation stability.

Model Selection Rationale: Although Random Forest (RF) and Decision Tree (DT) achieved comparable accuracy to XGBoost, the final selection of XGBoost was based on its superior F1 score, AUC, and cross-validation consistency. Its robustness, efficiency, and generalization ability make it well-suited for real-time healthcare scenarios.

Figure 4.12 shows the accuracy of all models, with XGBoost achieving the highest at 99.06%. Figure 4.13 presents confusion matrices that reveal classification performance across all models.

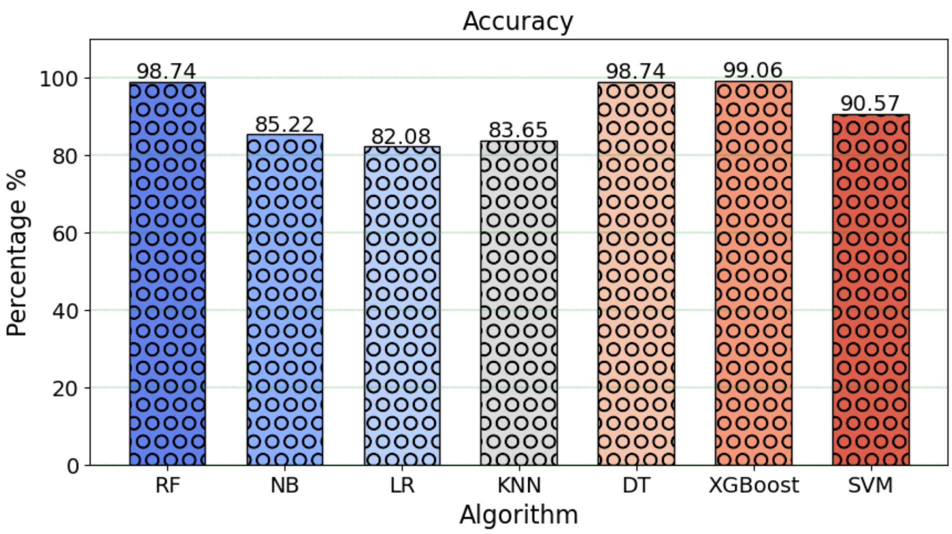


Figure 4.12. Comparison of model accuracy.

Figure 4.14 illustrates the performance in terms of precision, recall, and F1 score. XGBoost consistently outperformed others, especially in recall and F1 score, which are critical in healthcare to avoid false negatives.

The ROC and AUC analysis (Figure 4.15) further highlights XGBoost's classification power. Computation time (Figure 4.16) showed minimal overhead for XGBoost, confirming its feasibility for real-time tasks.

Table 4.2 summarizes cross-validation accuracy (cv=20). XGBoost achieved the best stability with the highest CVA (99.58%) and lowest standard deviation.

4.7.2 Deployment and Real-Time Prediction

Following model selection, XGBoost was deployed using Flask on a cloud-based platform with support for real-time predictions and alerting. Figure 4.17 shows API testing via Postman, where the deployed model delivered accurate predictions with 98% real-time accuracy.

This deployment confirms the model's practical value and readiness for real-time patient monitoring within a Digital Twin ecosystem.

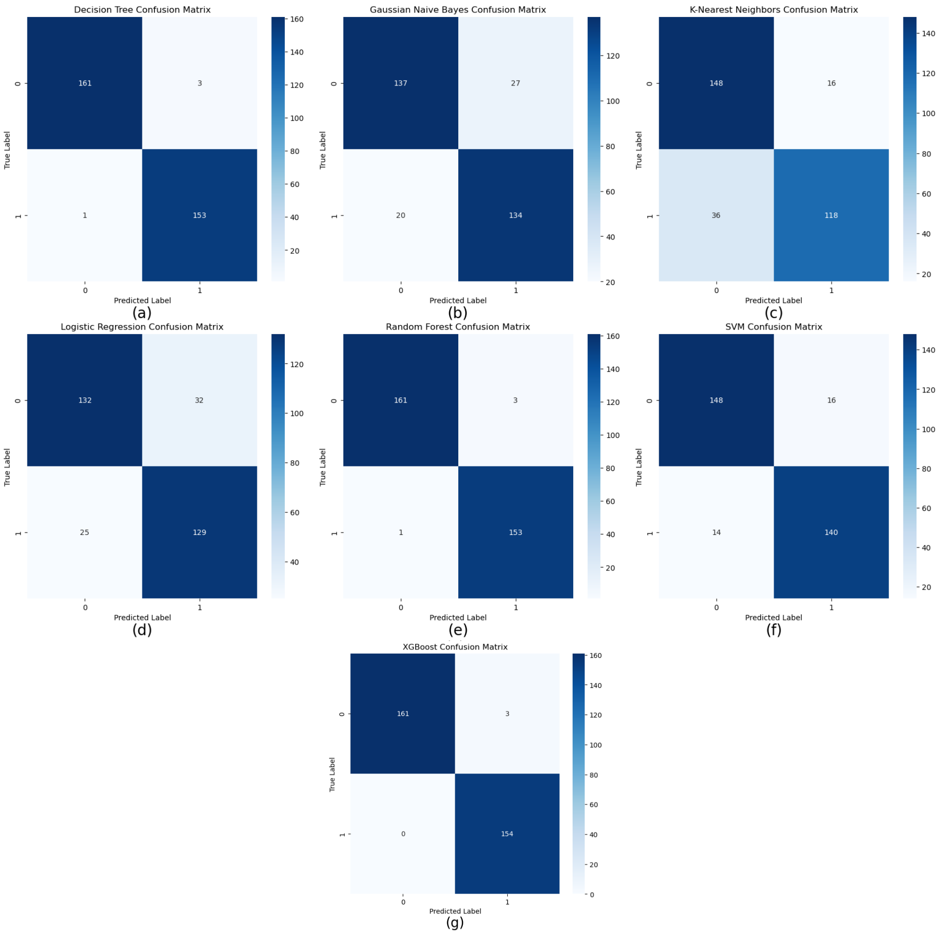


Figure 4.13. **Confusion Matrices of evaluated classifiers.**

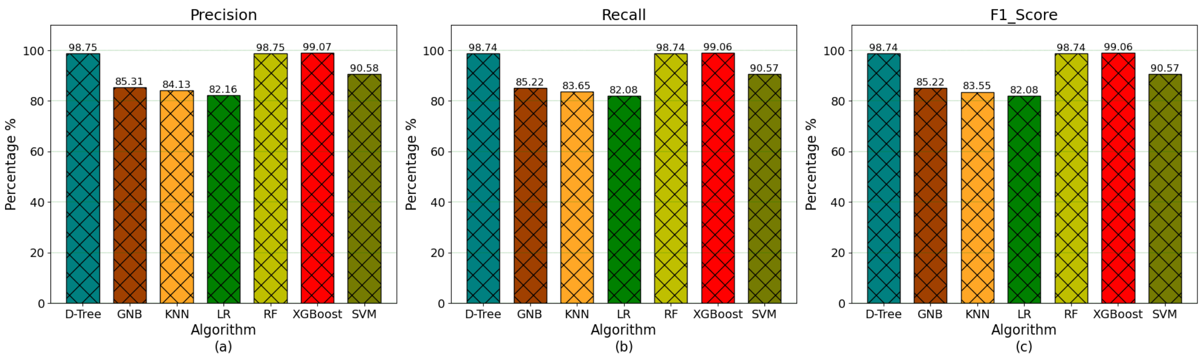


Figure 4.14. **Precision, Recall, and F1 Score comparison.**

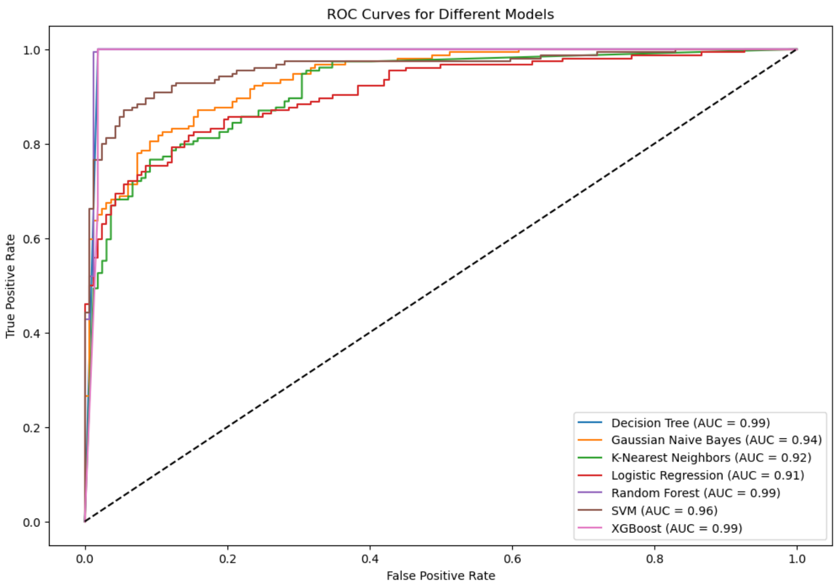


Figure 4.15. **ROC and AUC analysis.**

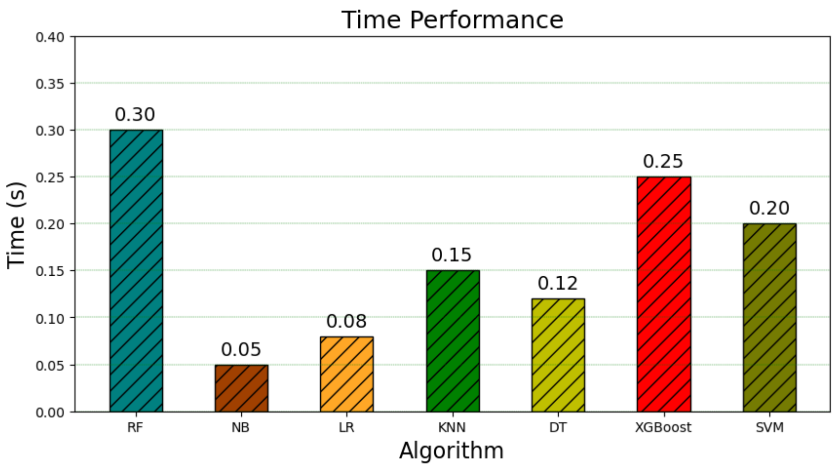


Figure 4.16. **Model computation time.**

Table 4.2. Cross-validation results (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

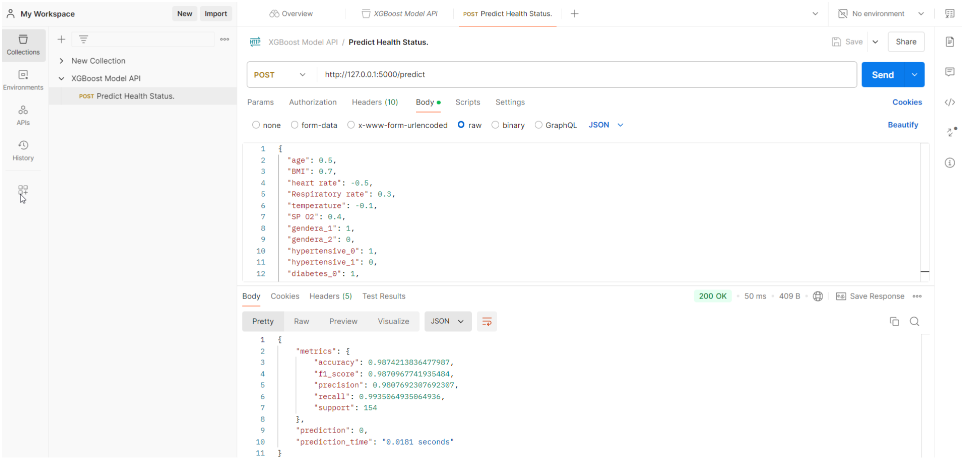


Figure 4.17. API Deployment Testing for Real-Time Prediction.

4.7.3 Web Portal

The web application was developed to visualize patient data and facilitate monitoring by healthcare providers and caregivers. Its core function is to integrate a Digital Twin (DT) using cloud infrastructure, Azure Digital Twin (ADT), and machine learning (ML) models for real-time prediction and alerting.

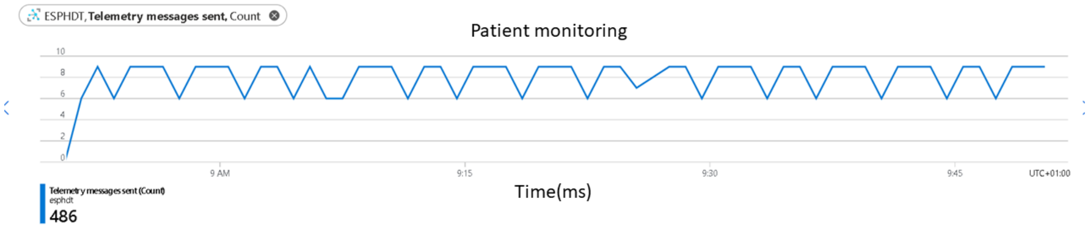
The portal supports data ingestion from wearable sensors, displays both physical and digital representations of vital signs, and triggers notifications (e.g., SMS or email) for abnormal conditions. The cloud backend manages data storage, stream analytics, and application logic to ensure seamless updates and reliable operation.

As shown in Figure 4.18, the system was tested successfully, uploading sensor readings to the cloud and enabling predictive diagnostics. The digital model was constructed using sensor specifications, historical data, and real-time telemetry. Through Time Series Insights (TSI), physical and digital metrics were compared, showing high fidelity and accurate DT synchronization.

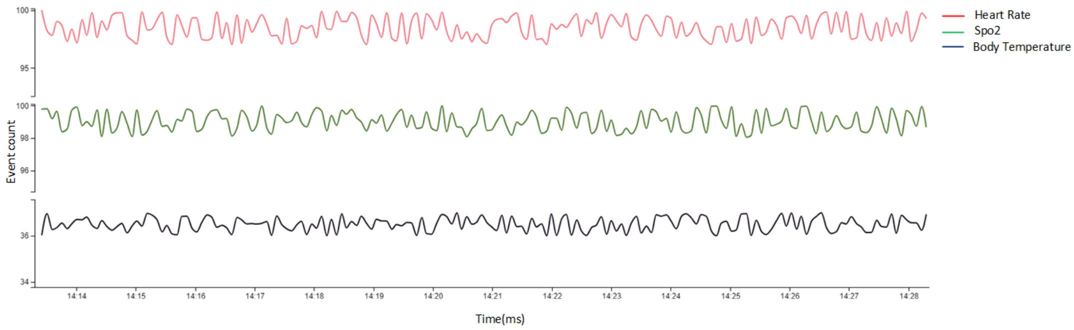
4.7.4 Model Performance Analysis

The Figure 4.19 shown the comparison of telemetry transmission latency for our model. It is observed that the latency increases steadily over time both cases. However, with Pyomo model optimised latency, maintaining it around 20-25 milliseconds, while without Pyomo, latency peaks at 40 milliseconds. This improvement demonstrated integrating Pyomo model achieved a 32% reduction in telemetry transmission latency.

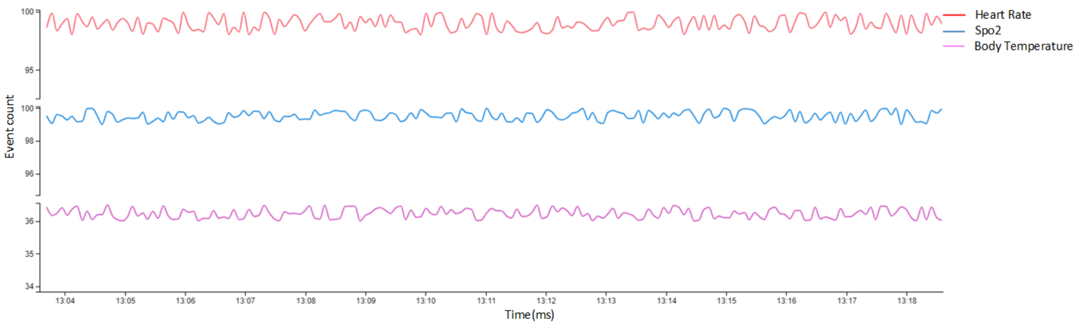
Figure 4.20 presents 3D comparison of DT system performance with and without Pyomo



(a) Prototype testing.



(b) Physical object (Sensors).



(c) Digital twin implementation.

Figure 4.18. Monitoring data (HR, SpO2, and BT) accessing the cloud server using the dashboard on the cloud in TSI: (a) Prototype testing, (b) Physical object (Sensors), (c) Digital twin implementation.

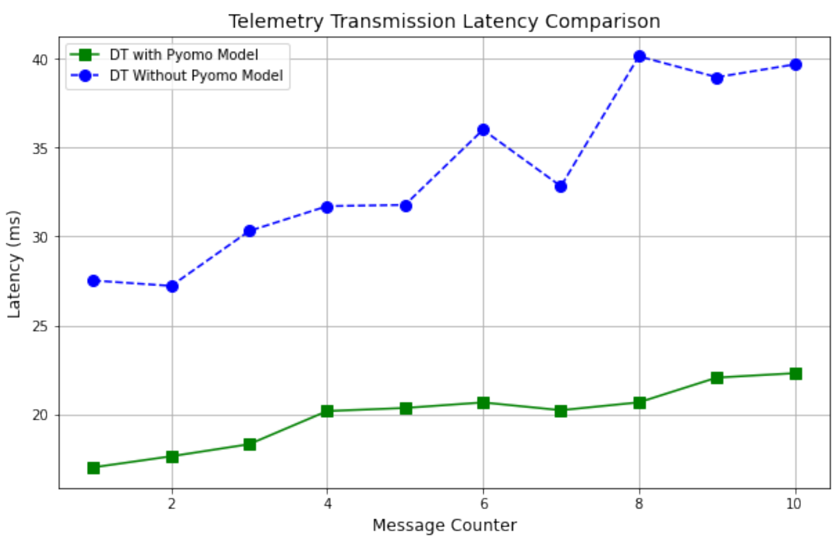


Figure 4.19. Comparison of Transmission Latency Over Time.

optimisation. The three axes represent:

1. X-axis: Runtime in seconds (how long the system runs during telemetry tasks).
2. Y-axis: Amount of data sent in bytes.
3. Z-axis: CPU usage percentage during that runtime.

The red line traces the performance of the system without Pyomo, while the blue line traces performance with Pyomo. Each point on the lines corresponds to one measurement of system behavior. The goal is to compare how resource usage and efficiency evolve over time for the two setups. For example, at a runtime of around 35 seconds, the DT system without Pyomo (red line) sends approximately 220,000 bytes but experiences a sharp CPU usage spike above 42%. In contrast, the DT with Pyomo (blue line) at the same data volume maintains a more stable CPU usage of around 30%, demonstrating better efficiency. This stability, visible through the smoother blue line, confirms that Pyomo helps to intelligently manage data flow and CPU usage under changing loads. It adapts telemetry frequency in real time, which is particularly valuable in healthcare where system reliability and low-latency responses are critical.

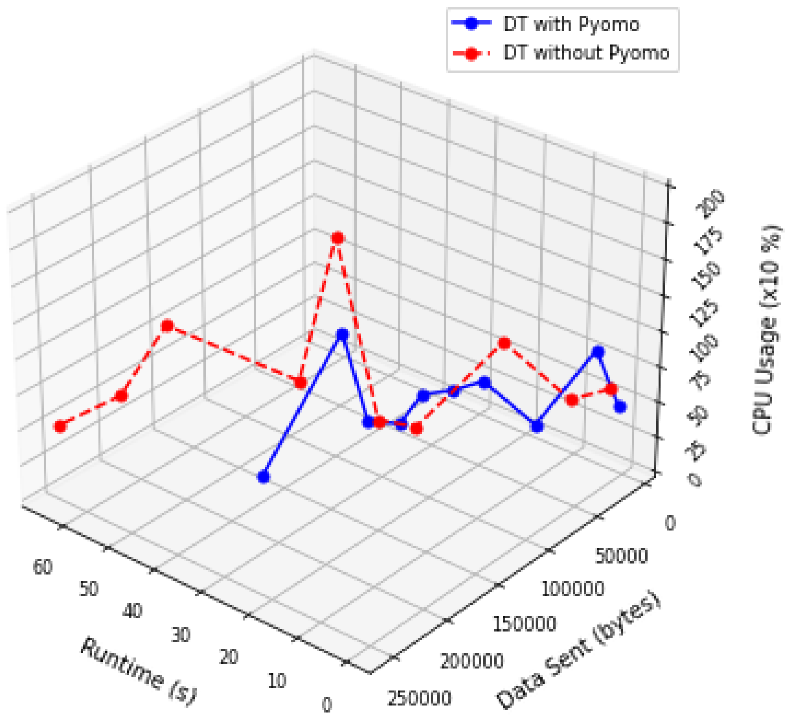


Figure 4.20. Performance Comparison of Runtime, Data Transmission, and CPU Usage for Digital Twin Systems with and without Pyomo.

The Figure 4.21 other results showed an improvement in system response time when the proposed model is incorporated. As seen from the presented model, gains in optimization lead to a nearly 52% improvement in response time improving the capability of the system to handle and respond to data.

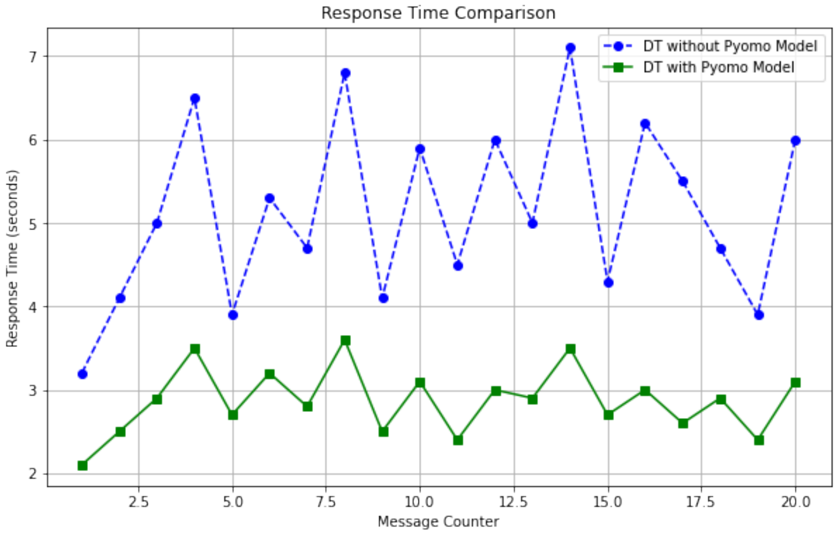


Figure 4.21. Response Time Comparison.

4.8 Conclusion and Summary

This study developed a system prototype utilizing DT methodology, IoT, ML, and AI techniques to enhance data interaction and integration within healthcare. This approach enables intelligent monitoring of physiological parameters such as HR, SpO2, and BT. The implementation of DT in healthcare can greatly support cloud-based services for elderly individuals and those with chronic medical conditions. The system is integrated with a graphical user interface based on ADT in real-time, allowing clinicians and patients to manage or monitor health effectively.

The wearable prototype is designed to be lighter, smaller, and more cost-effective, facilitating the monitoring of patients' vital signs. By employing edge computing methodologies, the system provides prompt and reliable local assessments while mitigating latency and detecting anomalous situations. The portability and wireless nature of the device enhance its ease of relocation. Machine learning was utilized to develop predictive models and process data, achieving 98% accuracy and 99.3% precision in real time using the XGBoost algorithm.

Chapter 5

Hybrid Cloud-Edge Digital Twin System with Quantum-Secured Real-Time Healthcare Monitoring

5.1 Introduction

In the context of Industry 4.0, which involves the convergence of advanced computational and communication technologies, the relevance of DTs in healthcare continues to grow. Several key challenges must be addressed to fully harness their potential, i.e., reducing latency in real-time monitoring, enabling personalised health management, and ensuring the integrity and security of healthcare data. These issues require a comprehensive approach that integrates cloud computing, edge computing, AI, and advanced security mechanisms, e.g., quantum-resistant cryptography and Quantum Key Distribution (QKD) [166]–[171].

In response to these challenges, the current chapter proposes a novel framework for DT in healthcare, contributing the following key innovations:

1. A patient-centric framework that leverages edge computing to reduce latency in healthcare data processing while adapting to individual patient needs for real-time monitoring and predictive analytics. This includes the QDTHS Algorithm, which enhances data transmission efficiency and security.
2. Enhanced data integrity and privacy within the DT framework by integrating quantum security mechanisms, i.e., Quantum Key Distribution (QKD), into the DTHQ(A,B,Q) protocol, ensuring health data protection against threats posed by classical and quantum computing advancements.
3. The application of AI-driven predictive models for accurate health metrics analysis, using

multi-dimensional techniques to track correlations between various health indicators and enable early detection of potential health issues. The QHIM Algorithm supports resource allocation and real-time predictive analytics.

4. Optimised scalability and resource management through cloud computing solutions that dynamically adjust to healthcare data demands, ensuring reliable system performance and minimising operational costs.

This chapter is organised as follows: Section 5.2 details the proposed DT model architecture. Section 5.3 explains the integration of quantum security into healthcare systems. Section 5.4 presents security evaluation and verification. Section 5.5 covers performance metrics and analysis, and Section 5.6 discusses the results. Finally, Section 5.7 concludes the chapter.

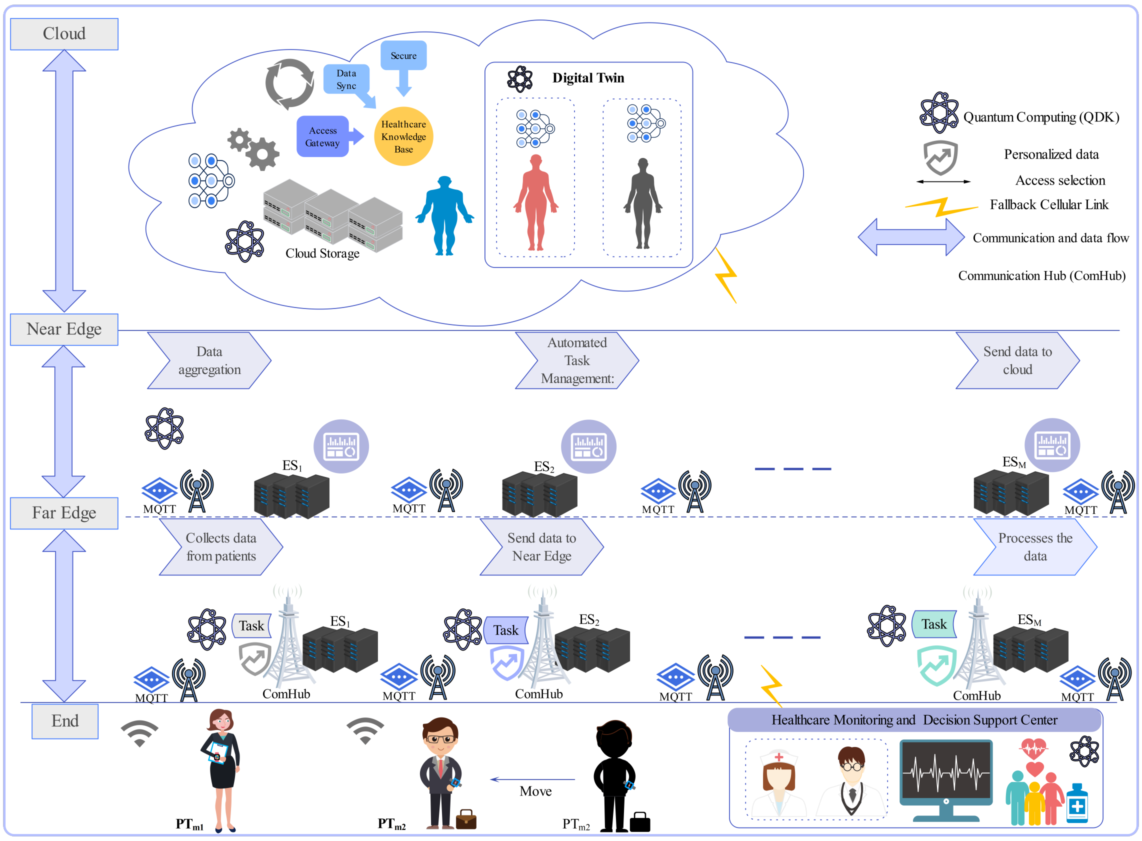


Figure 5.1. Proposed digital twin healthcare system architecture with secure data transmission in real time processing.

5.2 Proposed DT Model Architecture

5.2.1 Problem Formulation

The integration of advanced technologies, i.e., DT, IoT devices, and cloud computing, promises to enhance healthcare through improved patient monitoring and real-time interventions. However, significant challenges, such as scalability, security, and real-time data processing, must be addressed to fully realize these systems. Key challenges include:

1. Scalability of Healthcare Monitoring Systems: Traditional healthcare systems face difficulties in scaling to manage large volumes of real-time data from multiple patients. A scalable architecture is required to efficiently handle high data volumes while ensuring continuous and reliable monitoring.
2. Data Integrity and Security: Given the sensitivity of healthcare data, the advent of quantum computing poses threats to traditional encryption methods (e.g., RSA, AES). The aim is to implement quantum-resistant cryptography, such as QKD, to secure healthcare data transmission and storage.
3. Real-Time Data Processing and Predictive Analysis: Current systems experience latency and limited predictive accuracy, which hinder timely medical decision-making. An AI-driven architecture is proposed to enable real-time processing and predictive analytics, ensuring proactive healthcare interventions.
4. Interoperability and Integration: A lack of seamless integration between existing healthcare systems and emerging technologies (e.g., DT, IoT) must be addressed. The architecture should be flexible and interoperable to integrate smoothly with current healthcare infrastructure.
5. Resource Optimisation: Real-time data processing demands optimised resource management to handle the computational load. Cloud computing will be leveraged to dynamically allocate resources, ensuring cost-effective and reliable performance.

This research develops a healthcare monitoring architecture to address these challenges, enhancing system performance and improving healthcare outcomes.

5.2.2 System Overview

Figure (5.1) proposed Digital Twin healthcare system architecture, designed for scalability, real-time processing, and enhanced security. Data are collected by far-edge devices and processed at the near-edge layer for validation before transmission to the cloud. Automated task management and QKD ensure secure data flow. The cloud handles advanced analytics, storage, and updates to the digital twin, while IoT devices, edge servers, and communication hubs support real-time monitoring and decision-making.

Key components include IoT devices for data acquisition, secure data transmission using quantum-resistant techniques, and cloud-based AI models for predictive analysis. Real-time data flow between IoT devices, cloud services, and the digital twin model is expressed as

$$D_C(t) = f(S_r(t)) \rightarrow B_{\text{IoT}}(t), \tag{5.1}$$

where $D_C(t)$ represents the data collected at time t, $S_r(t)$ denotes sensor readings, and $B_{\text{IoT}}(t)$ signifies the IoT data buffer.

5.2.3 IoT Devices and Data Acquisition

The system relies on a suite of IoT devices equipped with precise sensors:

- MAX30102: Measures heart rate (HR) and blood oxygen saturation (SpO2).
- MLX90614: Measures body temperature (BT) through infrared sensing.
- NodeMCU ESP8266 Microcontroller: Acts as the central processing unit, handling sensor data and communication protocols.

Data collected by these devices is transmitted to the cloud using quantum-resistant encryption protocols via the MQTT protocol. The secure transmission process is defined as:

$$C_{\text{data}}(t) = E_{\text{QKD}}(P_{\text{data}}(t)), \tag{5.2}$$

where $C_{\text{data}}(t)$ represents encrypted data at time t, and $(P_{\text{data}}(t))$ refers to the plain data being transmitted.

5.2.4 Cloud Computing Infrastructure

The cloud infrastructure, built on Microsoft Azure, manages data storage, processing, and AI-driven predictive analytics. The Azure IoT Hub facilitates real-time communication between IoT devices and the cloud. The efficiency of data transmission is modeled as:

$$\mathcal{R}_{DT} = \frac{\mathcal{D}_T}{\Theta_{trans}}, \tag{5.3}$$

where $\mathcal{R}_{DT}$ represents the data transmission rate, $\mathcal{D}_T$ denotes the total data volume being transmitted, and Θ_{trans} corresponds to the transmission time.

All transmitted and stored data are secured using quantum-resistant encryption techniques, ensuring that patient information remains protected against classical and quantum-based cyber threats.

5.2.5 AI Prediction and Analysis Module

The AI-driven predictive analytics module processes real-time data from IoT devices, generating health predictions based on machine learning algorithms trained on both historical and real-time data. The predictive model function is expressed as:

$$\mathcal{P}(t) = f(\mathcal{B}_{IoT}(t), \mathcal{H}(t-1)), \tag{5.4}$$

where $\mathcal{P}(t)$ represents the predicted health parameters, $\mathcal{B}_{IoT}(t)$ denotes current sensor data at t, $\mathcal{H}(t-1)$ refers to the historical health data up to $(t-1)$ refers to historical health data.

To further enhance security, QKD ensures that sensor data are encrypted before being processed. The secure prediction model is expressed as:

$$\mathcal{P}_{QKD}(t) = f(E_{QKD}(\mathcal{B}_{IoT}(t)), \mathcal{H}(t-1)), \tag{5.5}$$

where $E_{QKD}(\mathcal{B}_{IoT}(t))$ represents the sensor data encrypted using QKD.

5.2.6 Security Mechanisms

The system applied QKD and quantum-resistant cryptography to secure communications and data storage within DT healthcare architecture. Figure (5.2) depicted the overall architecture for quantum encryption in the Digital Twin healthcare system, incorporating hybrid cryptography with QKD and AES-256 in CFB mode. The system integrates both far edge and near edge devices with lz4 and brotli compression, respectively, facilitating secure and efficient data transmission across the healthcare framework.

The process for securing data transmission, processing, and storage using these methods is demonstrated in Algorithm (5.1); Quantum Digital Twin Healthcare Security (QDTHS). The randomness generated by quantum circuits is crucial for secure key generation. As shown in Figure (5.6) (d), the quantum circuit results exhibit an equal distribution of measurement outcomes (0 and 1), ensuring the unpredictability required for quantum encryption.

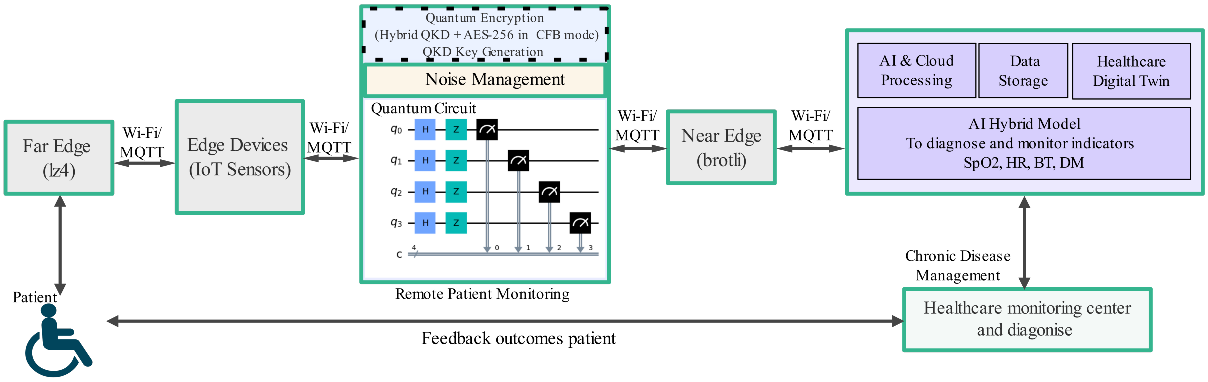


Figure 5.2. Quantum security circuit for healthcare data transmission: Integration of QKD with edge devices and AI analytics.

Algorithm 5.1. QDTHS Algoritm

Input: $\omega_{\text{conn}}(\text{ssid}, \text{password})$, Device Key, QKD_{key}
Output: $\mathcal{M}_{\text{Encrypted Data}}, \mathcal{R}_{\text{API}}, \mathcal{S}_{\text{Dashboard}}$
procedure $\mathcal{I}_{\text{init}}$ — # Initialise the system and secure connection
  $\omega_{\text{conn}}(\text{ssid}, \text{password}, \text{Device Key})$ — # Connect to WiFi and authenticate
  $\mathcal{QKD}_{\text{establish}}(QKD_{\text{key}})$ — # Establish a quantum key for secure communication
end procedure
procedure λ_{ML} — # Main Loop for data acquisition and processing
while τ **do**
  if Θ_{ST} **then** — # Time to send telemetry
    $d \leftarrow \alpha_{\text{CSD}}$ — # Collect sensor data
    $\delta_{\text{AZH}}(d, QKD_{\text{key}})$ — # Encrypt and send data using QKD
  end if
  $\xi_{\text{MCL}}()$ — # MQTT Client Loop for continuous communication
end while
end procedure
procedure $\delta_{\text{AZH}}(d, QKD_{\text{key}})$
  $\lambda_{\text{PDJ}}(d)$ — # Pack data to JSON format
  $\mathcal{M}_{\text{Encrypted Data}} \leftarrow \lambda_{\text{Encrypt}}(QKD_{\text{key}}, \lambda_{\text{PDJ}}(d))$ — #Encrypt data with quantum key
  $\mathcal{R}_{\text{API}} \leftarrow \mathcal{M}_{\text{Encrypted Data}}$ — # Send encrypted data to API Management
  $\mathcal{S}_{\text{Dashboard}} \leftarrow \mathcal{M}_{\text{Encrypted Data}}$ — # Update Dashboard
  $\mathcal{M}_{\text{DT}} \leftarrow \mathcal{M}_{\text{Encrypted Data}}$ — # Update Digital Twin Model
end procedure

5.2.7 Real-time Data Processing and Monitoring

The system's real-time data processing engine provides healthcare professionals with dynamic visualisations and alerts when critical thresholds are reached. The secure real-time data processing ensures patient privacy, and all transmissions are encrypted using quantum-resistant methods.

5.2.8 Data Storage and Management

The cloud-based storage system uses encrypted storage methods to protect patient data. The data storage function is defined as:

$$\mathcal{S}_t = h(\mathcal{D}_B(t)), \tag{5.6}$$

where $\mathcal{S}_t$ is the secure storage of data at time t, $\mathcal{D}_B(t)$ data stored in other components of the system at time t, and h is the storage function that organizes and archives the data appropriately. The secure storage model with QKD integration is expressed as:

$$\mathcal{S}_{QKD}(t) = h(E_{QKD}(\mathcal{D}_B(t))), \tag{5.7}$$

where E_{QKD} encrypts data before storage. The storage system updates AI models using historical data, continuously improving prediction accuracy through an iterative learning process:

$$F(t) : \mathcal{E}(t) \rightarrow \mathcal{M}_{DT}(t+1), \tag{5.8}$$

where $F(t)$ represents the model update function, enhancing DT predictive capabilities based on feedback from past predictions.

The proposed DT architecture incorporates IoT devices, cloud computing, and AI-driven predictive analytics, all secured using quantum-resistant encryption techniques. The integration of QKD ensures that patient data remains protected against current and future quantum-based threats. This architecture offers a scalable, secure, and real-time solution for healthcare monitoring, significantly improving patient outcomes through proactive and informed medical interventions.

5.3 Quantum Security in Digital Twin Healthcare Systems

As DT frameworks become increasingly prevalent in healthcare for real-time patient monitoring, the protection of sensitive patient data must be prioritised. Traditional encryption methods, such as RSA and AES, are increasingly vulnerable to emerging quantum computing capabilities (e.g., Shor's algorithm). This has accelerated the need for quantum-resistant security measures, particularly in environments requiring low latency and high reliability.

While blockchain has been widely explored in healthcare for data security, its reliance on decentralised consensus mechanisms introduces latency and computational overhead making it less suited to time-sensitive Digital Twin Healthcare (DTH) systems. In contrast, our approach integrates QKD and post-quantum cryptography to ensure secure communication, forward secrecy, and provable eavesdropping detection without the need for distributed ledger management. This enables scalable, real-time, and secure data flow between IoT devices, edge layers, and cloud infrastructure.

5.3.1 Implementing Quantum Security in DT Healthcare

The proposed framework safeguards healthcare data through a hybrid security mechanism that integrates QKD and quantum-resistant algorithms. The key components include:

- **Secure Communication Channels:** QKD ensures that encryption keys are exchanged with quantum-level security, and any interception attempt is detectable.
- **Post-Quantum Encryption:** Data within the DT system is encrypted using algorithms resistant to both classical and quantum attacks, securing both storage and transmission.
- **Authentication and Replay Protection:** Nonce-based verification and symmetric key exchange prevent impersonation and replay attacks within the DTH framework.

These components ensure a balance between high performance and strong security, enhancing the DT framework's resilience to current and future cyber threats while maintaining compliance with real-time operational demands. Table (5.1), important requirements for secure data transfer and computation were identified, namely the strength of cryptographic protection, communication delay, and resistance to quantum noise.

5.4 Security Evaluation and Verification

The security of the DTHQ(A,B,Q) protocol has been assessed through both informal and formal evaluations. Informal analysis has identified the core security features, while formal verification has been performed using the Scyther Verification Tool to validate resilience against known attack vectors.

5.4.1 Informal Security Evaluation

The DTHQ(A,B,Q) scheme allows an exchange of a quantum key Kq which proceeds confidentiality and message authenticity by incorporating both traditional and post quantum cryptography. Key security mechanisms include:

- Confidentiality: Attributes such as sensitive data, for instance, Kq is protected using predefined SEC key of entities A and Q secKeyA so as to ward off unauthorised persons.
- Authentication: The steps of ensuring mutual authentication involve exchange of nonces between A and B, and also swapping of public keys using PKA encryption in such a way that only a recipient can decipher the messages and respond.
- QKD: Quantum are implemented, defending against quantum computing attacks. All the above mentioned functions are symmetric and the quantum keys are transmitted through an entity Q and help in preventing session issuing.
- Replay Protection: Nonce (nA, nB) protects against replay attack, as captured messages cannot help the adversary to pretend they are partners in the conversation.

5.4.2 Formal Security Verification

Using the Scyther verification tool, the DTHQ(A,B,Q) protocol was put through a formal verification. This automated tool affirmed the secret, authenticity, and message interop of the protocol and affirmed its immunity to the mentioned types of attacks. While the basic objective of the verification process was to discover possible weak links and to be better prepared for threats, which were by then known. Scyther results are presented in the Figure (5.3).

- Secrecy Claims: Scyther confirmed that secret values, i.e., nonces (nA, nB) and Kq, remained confidential and were not accessible to unauthorised entities.

- Alive Claims: Both entities A and B confirmed each other's engagement, thus guaranteeing the protocols' compliance with the communication process.
- Message Integrity: Data integrity checks were performed and the results were positive which indicated there was no interference on the communicated data.

As illustrated in Figure (5.3), all security claims were successfully validated, with no vulnerabilities such as man-in-the-middle, replay, or impersonation attacks identified.

Scyther results : autoverify

Claim				Status		Commen
DTHQ	A	DTHQ,A2	Secret secKeyA	Ok	Verified	No attacks.
		DTHQ,A3	Secret nB	Ok	Verified	No attacks.
		DTHQ,A4	Secret nA	Ok	Verified	No attacks.
		DTHQ,A5	Secret reqKey	Ok	Verified	No attacks.
		DTHQ,A6	Secret Kq	Ok	Verified	No attacks.
		DTHQ,A7	Alive	Ok	Verified	No attacks.
		DTHQ,A8	Weakagree	Ok	Verified	No attacks.
		DTHQ,A9	Niagree	Ok	Verified	No attacks.
		DTHQ,A10	Nisynch	Ok	Verified	No attacks.
	B	DTHQ,B2	Secret nB	Ok	Verified	No attacks.
		DTHQ,B3	Secret Kq	Ok	Verified	No attacks.
		DTHQ,B4	Secret nA	Ok	Verified	No attacks.
		DTHQ,B5	Alive	Ok	Verified	No attacks.
		DTHQ,B6	Weakagree	Ok	Verified	No attacks.
		DTHQ,B7	Niagree	Ok	Verified	No attacks.
		DTHQ,B8	Nisynch	Ok	Verified	No attacks.
	Q	DTHQ,Q2	Secret secKeyA	Ok	Verified	No attacks.
		DTHQ,Q3	Secret Kq	Ok	Verified	No attacks.
		DTHQ,Q4	Secret nA	Ok	Verified	No attacks.
		DTHQ,Q5	Secret reqKey	Ok	Verified	No attacks.
		DTHQ,Q6	Alive	Ok	Verified	No attacks.
		DTHQ,Q7	Weakagree	Ok	Verified	No attacks.
		DTHQ,Q8	Niagree	Ok	Verified	No attacks.
		DTHQ,Q9	Nisynch	Ok	Verified	No attacks.

Done.

Figure 5.3. Formal verification of DTHQ protocol using Scyther tool.

5.5 Performance Metrics and Analysis

5.5.1 Simulation Setting

The simulation environment was designed to ensure robust performance and accurate evaluation of the proposed algorithms. The system operates on the ESP-IDF framework, built on FreeRTOS, which enables real-time patient monitoring using digital twin technology on a cloud platform. Simulations were performed on a custom-built machine with Intel Core i7-9700K CPU (8 cores, 3.6 GHz) and NVIDIA GTX 1080 Ti GPU, providing the necessary computational capacity for executing algorithms and managing simulation workloads.

Data were acquired via IoT devices, which transmitted information to cloud. Communication utilised the MQTT protocol over a 2.4 GHz Wi-Fi network (i.e., IEEE 802.11n). Azure IoT Hub served as the secure gateway for transmitting data, while cloud services (e.g., API Management and IoT Hub) were used for processing and storage.

The system operates within the Windows OS environment, with simulation algorithms implemented using Python 3.10.9. In the same way, the security of the protocol was checked using Scyther on Ubuntu 22.04 running via WSL2 on Windows. The sets of hardware and software guarantee to have an effective positive feedback fast, highly capable and secure simulation environment for timely monitoring of various patient data and effective testing of the digital twin security system.

Key parameters governing system operation, including data processing, resource scaling, and fault tolerance, are outlined in Table (5.1).

Table 5.1. Parameter Definitions for Digital Twin, AI, and Quantum Security in Healthcare

Parameter Name	Symbol	Type	Value	Description
Data Volume	ΔV	Integer	1 MB to 10 GB	Size of data processed or transmitted by the system, varies by application.
Data Block Size	-	Integer	1 KB, 4 KB, 16 KB	Size of data blocks for encryption, affects processing and transmission speed.
System Demand	ΔD	Integer	0 to 100%	Current load on the system as a percentage of total capacity.
Processing Adjustment	ϵ	Float	0.1 to 1.0	Adjustment step for scaling resources based on demand.
AI Model Complexity	C_{AI}	Integer	Low, Medium, High	Complexity level of the AI model, impacting computation load and prediction accuracy.
Energy Consumption	$\mathcal{E}_C$	Float	10W to 500W	Energy consumed during real-time processing and data transmission, measured in watts.
Real-Time Processing Latency	$\mathcal{L}_{RT}$	Integer/Float	100ms to 500ms	Latency in processing real-time data and delivering results, measured in milliseconds.
Threshold for Scaling (High)	Υ_{high}	Integer/Float	70% to 90%	Upper system load threshold for scaling resources.
Threshold for Scaling (Low)	Υ_{low}	Integer/Float	10% to 30%	Lower system load threshold for reducing resources.
Quantum Key Length	-	Integer	128, 256, 512 bits	Length of the quantum encryption key for data protection.
Quantum Noise Tolerance	-	Float	1% to 5%	Tolerance level for noise in the quantum communication channel.
Encryption Efficiency	$\mathcal{E}_{\text{enc}}$	Float	80% to 99%	Efficiency of encryption methods, i.e., quantum-resistant algorithms, in securing data.
Redundancy and Fault Tolerance	Σ_{RFT}	Integer/Float	1 to 5	Degree of system redundancy and fault tolerance.
Backup Service Status	Ψ	Boolean	True/False	Whether the backup service is active for fault tolerance.

5.5.2 Healthcare Infrastructure Integration

The DT framework has been meticulously crafted to harmoniously incorporate with prevailing healthcare infrastructures, facilitating secure and effective data exchange among systems such as electronic health records (EHR) and hospital information systems (HIS). The proposed integration guarantees alignment with established standards such as FHIR and HL7 in the administration of healthcare data. The security framework is fortified by employing Quantum Key Distribution (QKD), which safeguards confidential patient data during transmission and provides resilience against possible quantum computing threats. Patient information exchange and management require secure handling and transfers and are enhanced by the QHIM Algorithm (5.2).

Algorithm 5.2. QHIM Algorithm

```
Input: P_data, P_policy, P_id
Output: R_EHR, R_HIS, S_policy
procedure Establish_Shared_Access_Policy
    S_policy = NewPolicy(P_policy)
    SetPermissions(S_policy, Permissions)
end procedure
procedure Secure_Data_Transmission_to_EHR
    E_endpoint = GetEndpoint(P_id)
    R_EHR = SendData(E_endpoint, P_data, QKD)
end procedure
procedure Integrate_Data_with_HIS
    H_data = FormatForHIS(P_data)
    R_HIS = TransferToHIS(H_data, QKD)
end procedure
procedure Synchronise_Data_Between_Systems
    SyncData(R_EHR, R_HIS)
end procedure
        return R_EHR, R_HIS, S_policy
```

5.5.3 Scalability and Fault Tolerance

The system has been designed to autoscale cloud resources based on real-time requirements, thereby maintaining maximum efficiency and stability even during periods of high load. Resource scaling is tendered by means of predefined algorithms that control the intensity of resource usage to achieve objectives at the least expense. Business continuity or disaster recovery measures make data availability and data consistency possible in times of system loss or data damage. New policies of enhancing density of cloud computation, distribution of load in nodes and prevention of safe operations have been implemented in QHIM Algorithm (5.2). As such, active services to fail over are used only when necessary to ensure continuous operation, while at the same time cloud density is achieved as expected. Disaster recovery backup policies are used to get data back, and minimise lengthy time out of the system. The capacity of the system is maintained, and its ability to augment its service delivery based on existing demand is achieved.

5.5.4 AI-Driven Real-Time Health Monitoring and Analysis

To achieve reliable solution for constant patient care, predictive analytics based on AI is included in the framework of DT, combined with health monitoring in real time. Current and continuous IoT sensor data feed into an analysis engine and historical patient records to determine critical health factors needed for real-time diagnostics.

Key Features:

- Predictive Analytics: The data collected comes into the AI model which is hosted on the cloud to provide out predictive analysis. The illnesses are predicted in advance, and in case of occurrence, actual signals are given to doctors. As new data is assimilated the model updates its predictions thus increasing its accuracy and relevancy to clinical practice.

System Operation:

- Data Integration and Analysis: Information gathered from the IoT devices is sent to the DT model hosted in the cloud, where the advanced AI capabilities mirror the state of the patient with potential risks to health at an early stage.
- Immediate Alerts and Interventions: In the case where a pathology of abnormally changing conditions is recorded in the patient, signals are generated for healthcare workers, to act appropriately.
- Multi-Dimensional Analysis: General and specific health indicators are monitored by the DT system because the correlations between different aspects of health can be subtle, complex health conditions can be identified during the early stages.

5.5.5 Security-Related Performance

Using the proposed metrics of latency, communication cost, computational cost and throughput, DTHQ(A,B,Q) was assessed. Testing was performed in both far-edge and near-edge environments, with key lengths of 128, 256, and 512 bits. Compression algorithms, e.g., LZ4 and Brotli, were applied to simulate varying data sizes, offering insights into the protocol's efficiency within IoT environments, as presented in Table (5.2).

Table 5.2. Performance Metrics of DTHQ Protocol Across Edge Environments.

KL	ET	Data Size	CA	AL (ms)	CC (ms)	Thrpt (msgs/s)
128	Far Edge	Small	lz4	1.2	12	67.737
128	Near Edge	Small	brotli	4.2	45	3880.330
256	Far Edge	Medium	lz4	1.3	15	682.756
256	Near Edge	Medium	brotli	5.1	47	2039.359
512	Far Edge	Large	lz4	2.5	21	1597.010
512	Near Edge	Large	brotli	8.4	52	847.314

CC:Computational Cost; CA:Compression Algorithm;Thrpt:Throughput; AL: Average Latency;messages per second (msgs/s): Key Length(KL); Edge Type(ET).

Quantum-resistant cryptography is incorporated into the protocol, ensuring security against classical and quantum threats. LZ4 was selected for far-edge deployments to prioritise speed, while Brotli was employed in near-edge cases to optimise bandwidth. Latency was calculated based on the total time required to compress, encrypt, and transmit data, utilising Python's perf_counter() function. Communication cost reflected the number of bits transmitted post-compression, and computational cost was calculated by measuring the time spent on cryptographic operations. The use of Elliptic Curve Cryptography (ECC) for key exchange and symmetric encryption minimised computational overhead. The total computational cost is expressed as:

$$\text{Computational Cost} = 2T_{ECC} + 3T_H + T_E + T_D \tag{5.9}$$

Where:

- T_{ECC}: Time for Elliptic Curve Point Multiplication (for key exchange).
- T_H: Time for hashing operations.
- T_E: Time for symmetric encryption.
- T_D: Time for symmetric decryption.

These operations, by leveraging ECC instead of traditional modular exponentiation, significantly reduce computational overhead. Comparative protocols, such as those in [172] and [173], rely on more resource-intensive cryptographic formulas, which include modular exponentiation (e.g., 11TM + 7TH + TB). DTHQ's reliance on ECC allows it to achieve greater efficiency, especially in IoT environments with constrained resources.

The cryptographic components, such as a 256-bit symmetric encryption key and 160-bit hashing, were selected to strike a balance between security and performance. For example, during login, the total communication cost per message was 896 bits, which included components such as avatar identification, nonces, and timestamps.

The performance results, illustrated in Figure (5.4), show the following trends:

- Latency: Far-edge deployments exhibit higher latency, especially with smaller key lengths and data sizes Figure (5.4) (a).
- Throughput: Near-edge deployments achieve more consistent throughput with smaller data sizes, while far-edge environments handle larger data sizes more effectively Figure (5.4) (b).
- Computational Cost: Far-edge configurations show higher costs with smaller key lengths, whereas larger data sizes reduce costs across both environments Figure (5.4) (c).

- CPU Usage: Throughput increases steadily, despite fluctuations in CPU usage during peak system load Figure (5.4) (d).

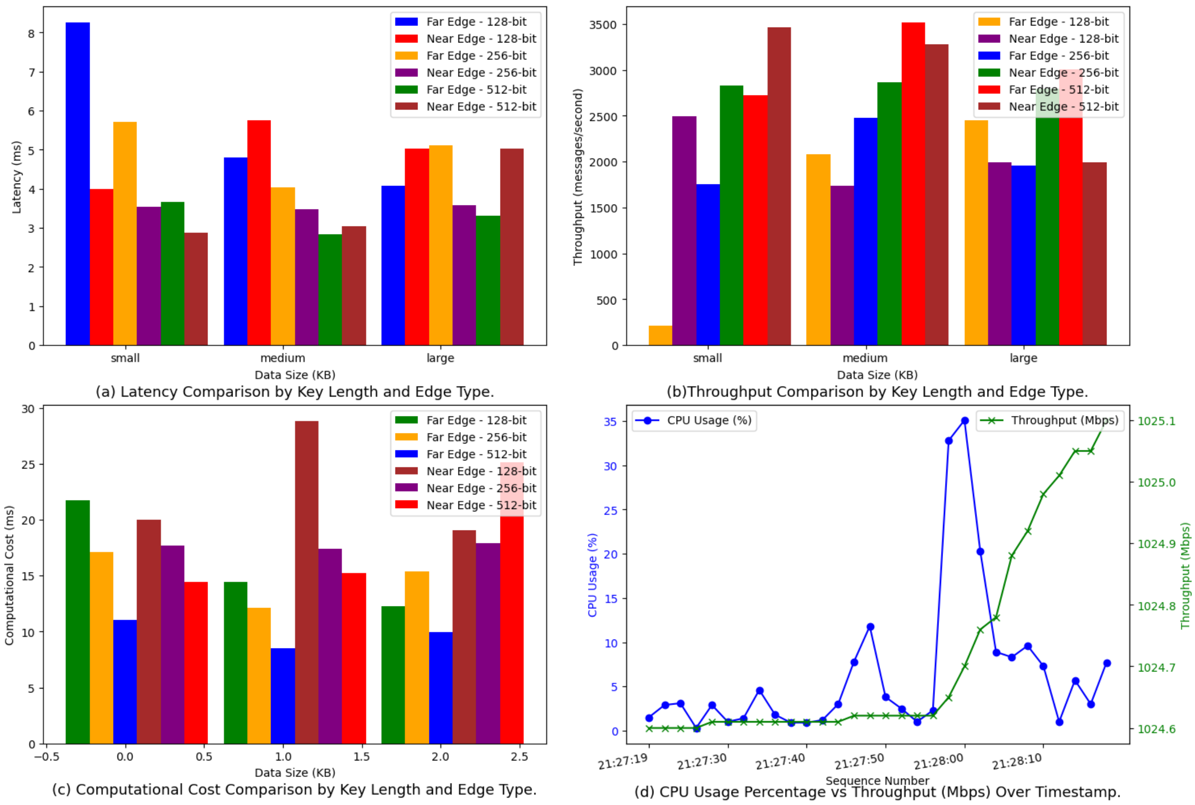


Figure 5.4. Performance Analysis of Edge Computing Based on Key Length, Edge Type, and Data Size.

5.5.6 Hybrid Model Algorithm

A hybrid model combining the Multilayer Perceptron (MLP) and Extreme Gradient Boosting (XGBoost) classifiers has been developed to balance computational efficiency and predictive accuracy for healthcare applications. This approach integrates the deep learning capabilities of MLP with XGBoost's decision trees to optimize performance across multiple target variables, as shown in Figure (5.5). The hybrid model achieves a 58.40% reduction in testing time compared to individual classifiers, making it a practical solution for real-time healthcare monitoring.

Data Loading and Preprocessing

The dataset, sourced from the MIMIC-III Public Health Dataset, includes 1,177 medical records and several critical features such as TargetHR, TargetSpO2, TargetBT, and TargetDM [174]. Advanced preprocessing techniques, including feature selection and imputation, were applied

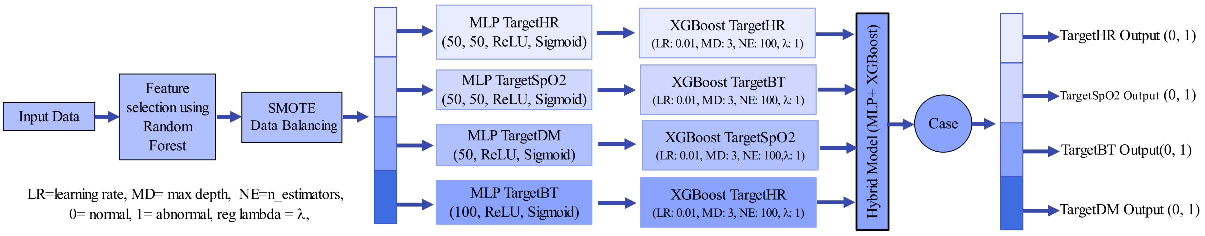


Figure 5.5. Hybrid model architecture.

to handle missing data. Gaussian noise was introduced to improve the robustness of the model, allowing it to generalize better under noisy conditions.

Model Training and Structure

The hybrid model's architecture was designed to maximize accuracy while minimizing computational cost:

- MLP Classifier: A fine-tuned MLP classifier with hidden layers (50-100 neurons) and L2 regularisation to prevent overfitting, which is critical for reliable performance in healthcare applications. Cross-validation was used to tune the learning rate and regularisation parameters as can be seen in Figure (5.5).
- XGBoost Classifier: To further increase accuracy by reducing the model complexity and improve computational performance a gradient boosting decision tree model was fine tuned to have the values max_depth = 3, n_estimator = 100, and reg_lambda = 1.0 as detailed in Figure (5.5).
- Hybrid Structure: The proposed hybrid approach combines the outputs of both classifiers by the ensemble technique that is based on finding the mode defined in Equation (5.10).

$$P_{\text{final}} = \text{Mode}(P_{\text{MLP}}, P_{\text{XGB}}) \tag{5.10}$$

Multi-target Classification

Designed for multi-target classification, the hybrid model focuses on healthcare indicators such as SpO2, body temperature, and diabetes mellitus (DM). Binary classification is applied to each target, with the ensemble method minimising misclassifications and ensuring more accurate predictions.

Testing Time and Efficiency

A decrease of testing time from MLP and XGBoost classifiers by 58.40% makes the hybrid model more suitable for real-time healthcare. There being no loopholes for redundancy to creep in, this optimisation augments functionality while at the same time shrinking the possibility of error.

Data Augmentation via GANs

To reduce the risk of overtraining due to small amount of data, Generative Adversarial Network was used to generate the data. The use of GANs broadened the training set resulting in enhanced generalisation capability of the hybrid model across different healthcare datasets.

- GAN Architecture: The GAN is comprised of the Generator which generates synthetic healthcare data and the Discriminator which determines their legitimacy. The model was trained for 10000 epochs; meanwhile Discriminator tries to tune the Generator to a level where Discriminator no longer able to differentiate the output of Generator and real data.

The synthetic data generated by the GAN was combined with the real dataset, improving the hybrid model's ability to handle imbalanced data and enhancing its overall performance.

5.6 Results and Discussions

5.6.1 System-Level Performance Evaluation

This section aims to validate the system-level performance improvements introduced by the proposed DTH framework. A comparative simulation was conducted to assess the effectiveness of DT integration across multiple performance metrics, including network latency, throughput, and operational efficiency. The evaluation covered a wide range of data sizes, from 1 KB to 100 MB, and demonstrates the advantages of the DT-driven architecture over conventional approaches.

As shown in Figure 5.6(a), the DT-assisted system achieved a 40% reduction in network latency. This improvement is attributed to the dynamic resource management mechanism implemented in Algorithm 5.2, which adapts processing power allocation according to system load and task priority, thereby minimizing communication delays and enhancing responsiveness.

In terms of throughput, Figure 5.6(b) illustrates a 30% gain compared to the baseline system. This enhancement results from the optimized patient data pipeline and real-time synchronization between IoT devices, edge layers, and cloud components. The unified management of digital twin entities enables faster processing and data delivery, contributing to more efficient health data handling and timely interventions.

Additionally, the DT-integrated system outperformed its non-DT counterpart by 15% in overall operational efficiency, as observed in Figure 5.6(c). This efficiency stems from the intelligent data acquisition and transfer strategy embedded in Algorithm 5.2, which reduces bandwidth usage, minimizes redundant computations, and ensures a balanced computational load across the infrastructure.

In summary, the results clearly indicate that the proposed DT framework significantly enhances system performance across critical dimensions. The deployment of task-aware optimization strategies and DT-based synchronization mechanisms demonstrates tangible gains in speed, data handling, and energy-aware computing, which are crucial for real-time healthcare monitoring and response.

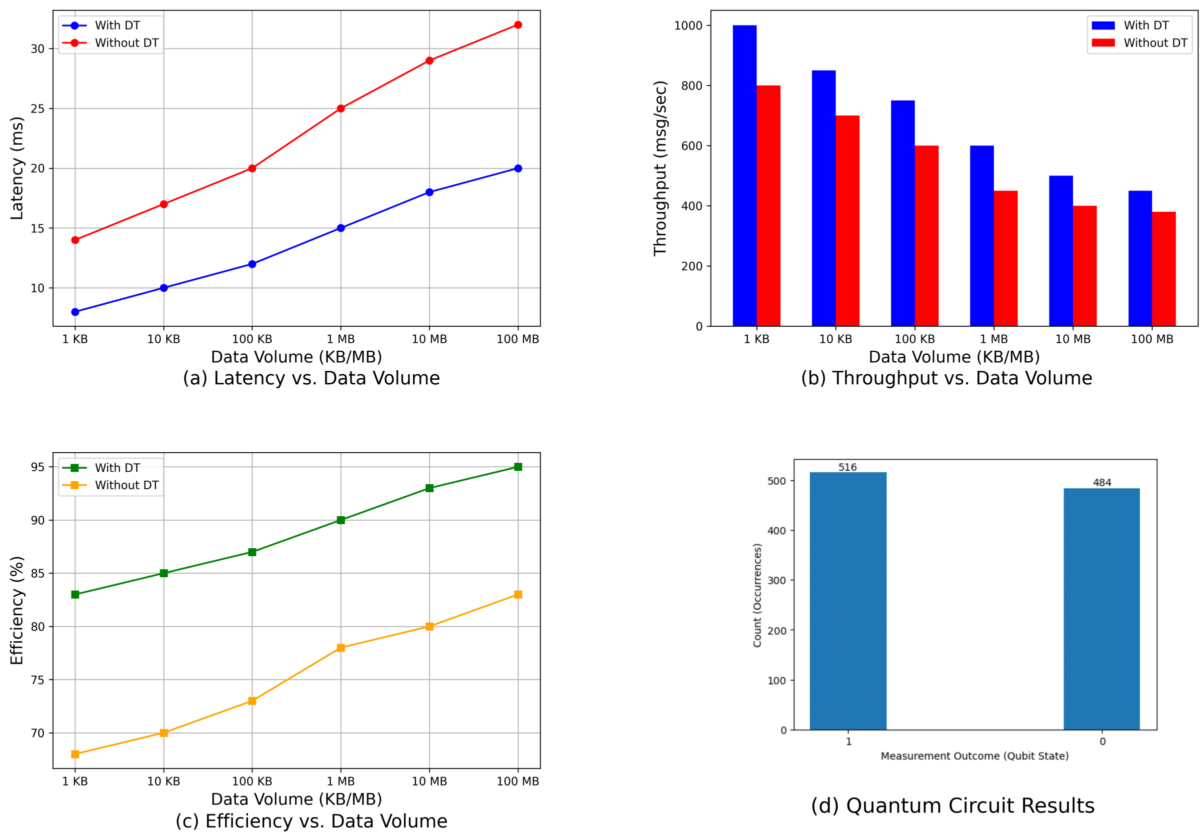


Figure 5.6. System Performance Metrics with and without Digital Twin Integration, and Quantum Circuit Measurement Results.

5.6.2 Hybrid AI Model Performance and Evaluation

The model was trained and validated using 10-fold cross-validation, and its performance was assessed against individual MLP and XGBoost models using accuracy, precision, recall, and F1-score metrics. Figure 5.7 presents macro and weighted average scores across all health indicators, while Table 5.3 offers a detailed breakdown. The hybrid model achieved the highest average accuracy of 97.48%, with corresponding precision, recall, and F1-score values of 95.27%, 97.57%, and 96.38%, respectively. These results demonstrate consistent performance improvements over both stand-alone classifiers.

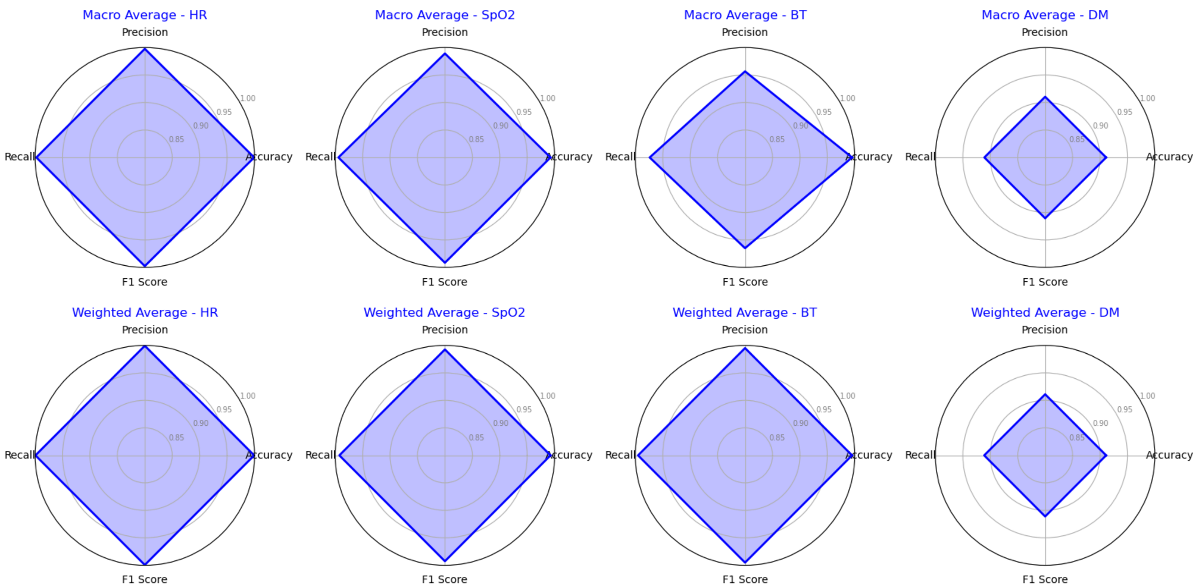


Figure 5.7. Macro and weighted averages for health metrics.

Table 5.3. Cross-Validation performance of Hybrid, MLP, and XGBoost models

Target	Model	Accuracy	Precision	Recall	F1 Score
HR	Hybrid	0.999	0.999	0.999	0.999
	MLP	0.999	1.000	0.997	0.998
	XGBoost	0.998	0.983	0.993	0.990
SpO2	Hybrid	0.997	0.999	0.997	0.997
	MLP	0.991	0.999	0.987	0.993
	XGBoost	0.989	0.995	0.988	0.992
BT	Hybrid	0.996	0.932	0.988	0.959
	MLP	0.996	0.942	0.958	0.950
	XGBoost	0.963	0.974	0.966	0.970
DM	Hybrid	0.905	0.880	0.918	0.899
	MLP	0.904	0.884	0.909	0.897
	XGBoost	0.903	0.880	0.913	0.896
Average	Hybrid	0.9748	0.9527	0.9757	0.9638
	MLP	0.9729	0.9567	0.9631	0.9598
	XGBoost	0.9636	0.8393	0.9666	0.8840

For interpretability, feature importance analysis from XGBoost (Figure 5.8) identified “diabetes” as the most influential feature across all targets. Other significant contributors included heart rate, glucose, and systolic blood pressure, while several features showed negligible impact suggesting redundancy that can be considered in future feature selection.

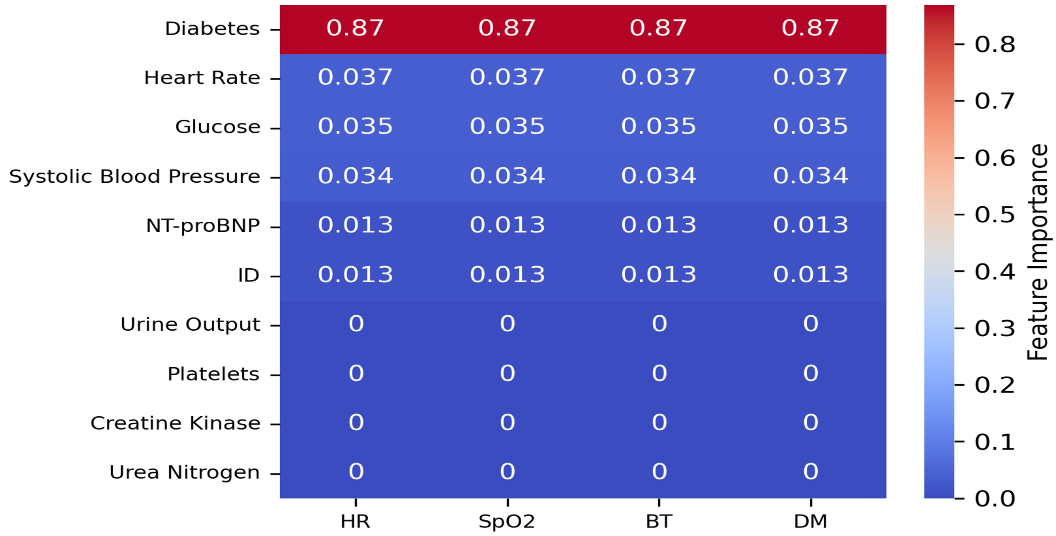


Figure 5.8. Feature importance scores across different targets (HR, SpO2, BT, DM) as determined by XGBoost.

To enhance robustness, Gaussian noise (mean = 0, std = 1.5) was added during preprocessing. This regularization step improved generalisation and maintained high accuracy under noisy conditions. Data leakage was carefully avoided through strict separation of training and testing sets.

Prediction errors were further analysed using distribution plots, rolling averages, and statistical residual metrics. Figure 5.9 illustrates consistency between predicted and actual values, while Figure 5.10 confirms low autocorrelation of residuals across time lags—indicating stable and unbiased predictions.

The autocorrelation of prediction residuals is presented in Figure 5.10, confirming that most residuals do not exhibit time-dependent structure. This analysis was extended using a multi-dimensional version of the autocorrelation function (ACF), which quantifies the time-lagged correlation between pairs of physiological variables. It is expressed as:

$$\rho_{X,Y}(k) = \frac{E\left[(X_t - \mu_X)(Y_{t-k} - \mu_Y)\right]}{\sigma_X \sigma_Y} \tag{5.11}$$

where X_t and Y_{t-k} denote two health metrics at time t and $t-k$, μ_X, μ_Y are their respective means, and σ_X, σ_Y are the standard deviations. This formulation helps identify inter-metric dependencies over time.

The Mean Absolute Deviation (MAD) under the highest noise level was 0.0709 (HR), 0.0961

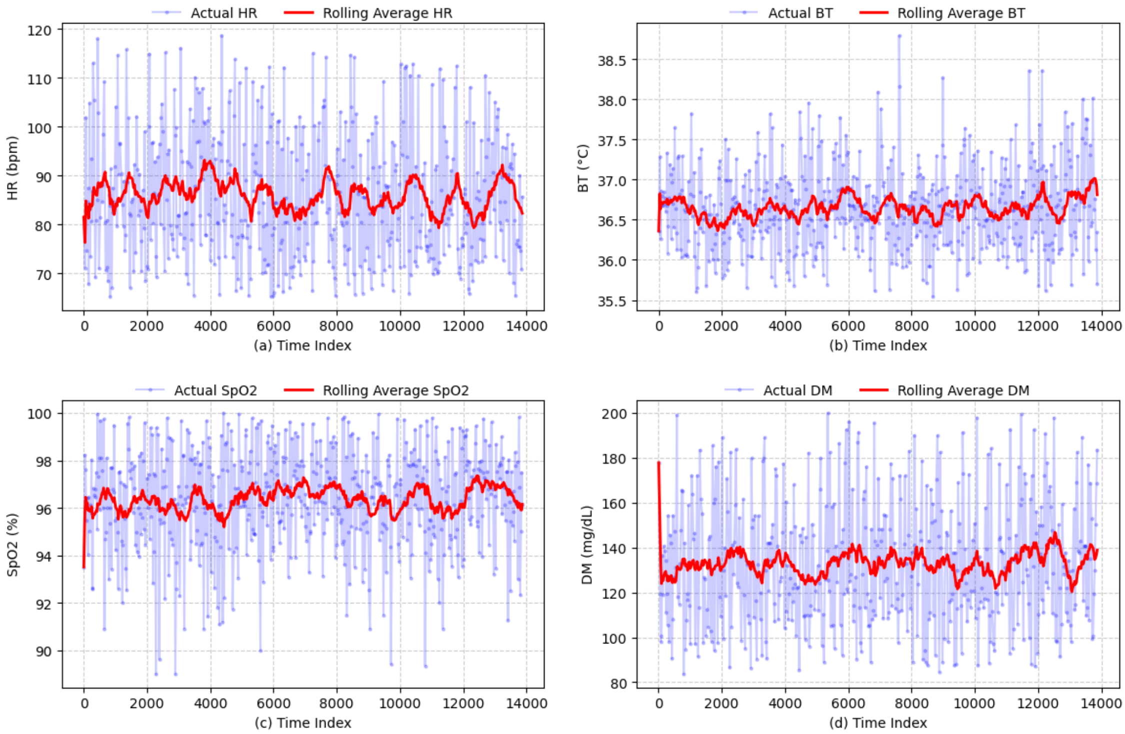
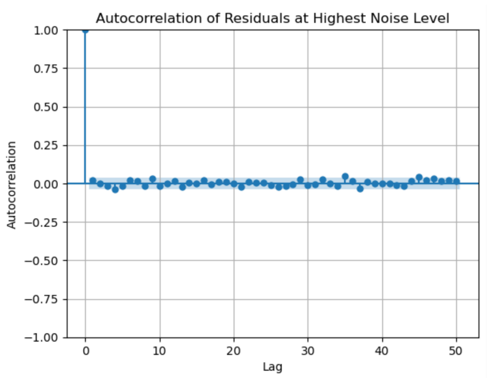
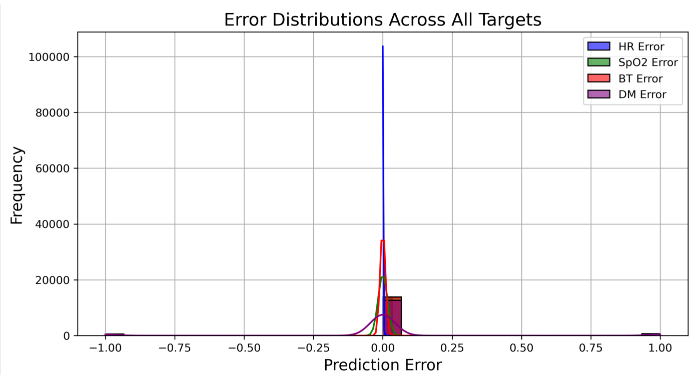


Figure 5.9. Comparison of actual vs. rolling average values for health metrics.



(a) Autocorrelation of prediction residuals at highest noise level.



(b) Error distributions across all targets.

Figure 5.10. Prediction error analysis and autocorrelation of residuals.

(SpO2), 0.0371 (BT), and 0.2560 (DM), confirming low error magnitude. Cross-correlation results in Figure 5.11 revealed weak but interpretable relationships, e.g., delayed correlation between HR and SpO2.

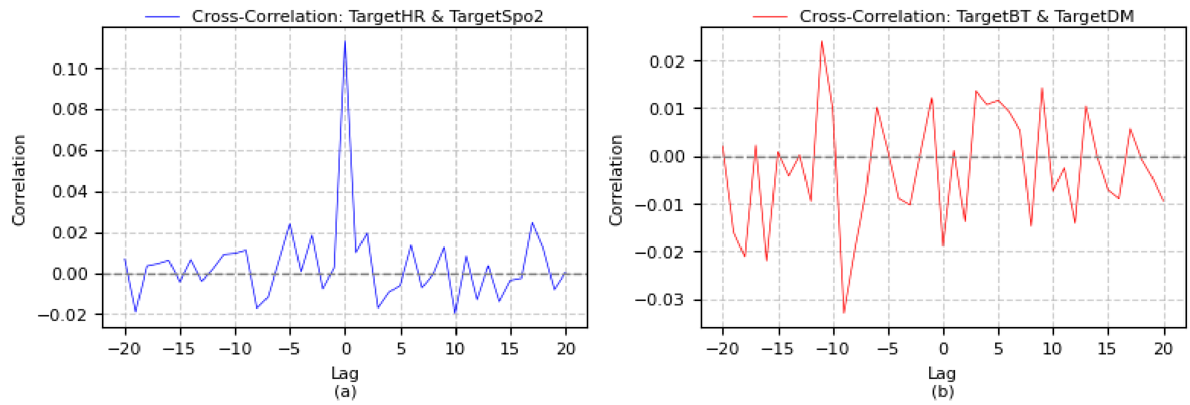


Figure 5.11. Cross-correlation analysis between health metrics.

Moreover, the Mean Absolute Deviation (MAD) was calculated to assess average prediction error under noise. The values obtained were 0.0709 (HR), 0.0961 (SpO2), 0.0371 (BT), and 0.2560 (DM). The MAD is defined mathematically as:

$$\text{MAD} = \frac{1}{N}\sum_{i=1}^{N} |\text{Residual}_i| , \tag{5.12}$$

where N is the number of predictions and Residual_i is the absolute difference between the predicted and actual value. A lower MAD indicates better prediction reliability across health indicators. Finally, to demonstrate deployment readiness, the trained hybrid model was served using a Flask-based REST API. The API accepts patient data in JSON format and returns predicted health metrics. Figure 5.12 shows a test request in Postman, validating the model's real-time performance. The system successfully handled concurrent requests with minimal latency, and the model remained consistent with cross-validation accuracy (95.45%) in live simulations.

Overall, the hybrid model exhibits strong predictive capability, high interpretability, robust performance under noise, and seamless deployment in real-time environments—supporting its practical applicability for intelligent digital twin healthcare systems.

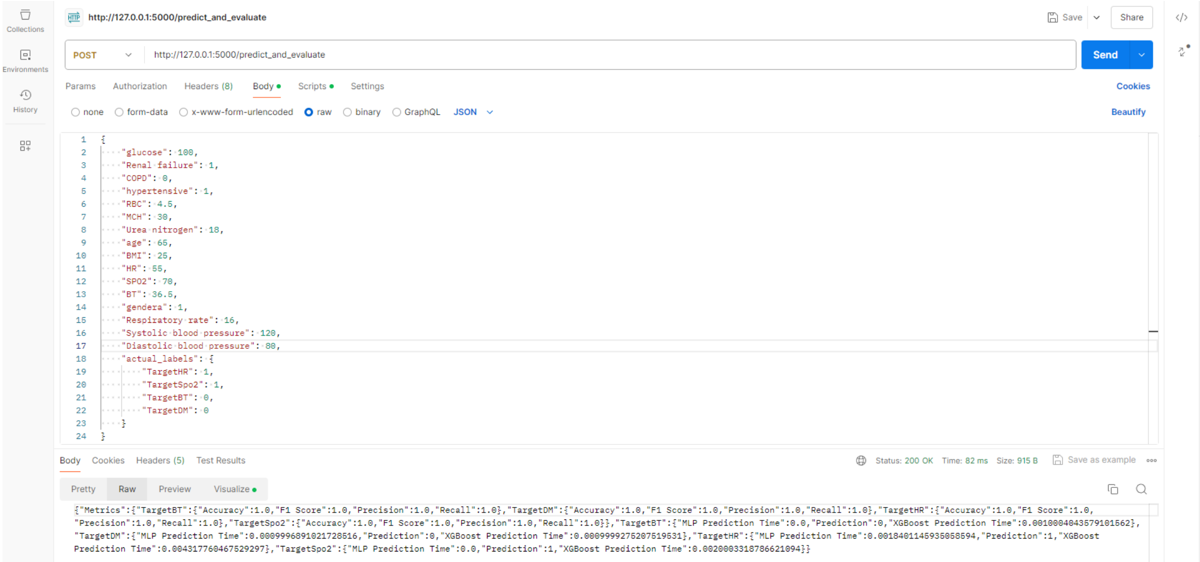


Figure 5.12. Postman interface showing prediction and evaluation results from the Flask API.

5.6.3 Quantum Computing Integration in Classical PC Systems: Challenges and Adaptations

Integrating quantum computing into traditional PC systems presents notable challenges due to the limitations of conventional hardware and software. Quantum computers, operating with qubits, differ significantly from classical systems, which process data in binary form. This disparity necessitates the use of specialized simulators, i.e., Qiskit or Microsoft's Quantum Development Kit, to emulate quantum behavior on classical hardware. However, these tools cannot fully replicate quantum computing's potential.

Quantum encryption, e.g., Quantum Key Distribution (QKD), further complicates integration, as traditional PCs require software modifications and network configurations. Adjustments, such as opening specific ports (i.e., port 443) and addressing quantum-specific errors (e.g., Error 402 and Error 442), are necessary. Despite these obstacles, simulators prepare developers for future quantum hardware deployment, while integrating quantum-resistant encryption within classical networks enhances data security. Through proper adaptation, traditional PCs can effectively support quantum technology exploration.

5.7 Conclusion

A hybrid cloud-edge quantum computing system has been presented in this study, designed to facilitate real-time healthcare monitoring. The system integrates DT technology, quantum security, AI-driven analytics, and IoT sensors. The hybrid AI model, combining MLP and XGBoost, achieved a cross-validation accuracy of 97.48% (CV = 10) and a real-time accuracy of 95.45% under noisy conditions. Additionally, a 58.40% reduction in testing time was observed compared to individual classifiers, enhancing computational efficiency for real-time applications.

The DTHQ(A,B,Q) protocol underwent rigorous security verification through the Scyther tool, confirming its resilience against various attack vectors. Quantum-resistant security measures were successfully applied to safeguard sensitive healthcare data, ensuring both privacy and integrity in anticipation of future quantum threats.

From computational standpoint, the hybrid model demonstrated a 25.77% improvement over prior work, while latency metrics reached 0.01 ms, ensuring near-instantaneous responses in real-time healthcare environments. The system was deployed via Flask, and validation in real-world settings confirmed its reliability, delivering consistent accuracy in healthcare predictions. Future research will aim to enhance system robustness by integrating more advanced AI models and quantum cryptographic techniques, further improving both predictive accuracy and security.

Chapter 6

Innovative Task Offloading Strategies in Healthcare: Integration of Digital Twins and Social Health Determinants

6.1 Introduction

Effective means of task offloading is particularly important in the context of healthcare disciplines as the improvement of response time and the availability of the health care services can significantly enhance the quality of the service. In recent years, DT technology has become one of the distinguishing technologies in achieving this objective. DTs generate real-virtual copies of physical objects so as to allow a simultaneous monitoring of the physical conditions, perform predictive analysis and provide personalised healthcare application. When used alongside operating modes of offloading tasks, the best practices of DTs will provide a sound framework for enhancing the efficiency and effectiveness of the delivery of care in health systems.

The proposed framework in this study presents a new concept of task offloading, particularly in the context of healthcare. The primary contributions of this study include:

1. Presenting a novel concept of integrating partial and binary offloading policies via an adaptive framework that selects the optimal mode based on task type and system conditions, as detailed in Algorithm 6.1, Section 6.2.
2. Leading the incorporation of DT and social health determinants into offloading discussions; promoting the primary precautionary stakeholders' health approaches and individualized patients' management strategies.
3. Providing the field-based real-world evidence of this knowing as the Digital Twin Healthcare Enhanced Asynchronous Team-Based Multi-Agent Proximal Policy Optimisation

(DTH-ATB-MAPPO), this study proves its effectiveness in comparison to the existing methodologies in terms of convergence of its rewards in the healthcare settings.

4. The study also integrates some newly identified aspects to extend the specification of DT adoption in refining MEC systems. It states that it achieves a percentage improvement of one sort or another, such as network delay and power usage.

5. Designing the experimental models that are well connected with the theoretical background and the practical applicability for conducting experimental research and backed on the basis of several cases' simulation data.

6. The ACTO algorithm has deployed the adaptive protection functions together with the exact matching technology to detect threats and provide adaptive cybersecurity.

7. In this work, a new algorithm named AI-Quantum-Digital Twin-IoT (AQDT-IoT) considers the quantum pre-processing to support decision-making regarding task offloading aiming both better performance and reliability.

The remainder of this study is structured as follows: The methodology is detailed in Section 6.2. In Section 6.3 outlines the proposed secure data offloading in healthcare. Section 6.4, the simulation setup and parameters are discussed. DT technology analysis with MEC system optimization is covered in 6.5. The results are discussed in Section 6.6. Finally, the study concludes in Section 6.7, summarising this study and discussing future research implications of our results.

6.2 Methodology

6.2.1 System Model and Framework

The core of this model comprises healthcare devices equipped with various critical biometric sensors and medical instruments that capture the data related to patient's health. This data is collected by edge computing units close to the new sources of such information so that localised data processing can be enabled. As a result, real-time analytics are enabled, reducing latency and improving the speed and responsiveness of the healthcare system. Subsequently, the collected health data is integrated into a virtual model, a fundamental component of DT technology. This virtual model is a detailed and dynamic software representation of the physical devices. Barometer readings, temperature, humidity, magnetometer readings, motion data, BT, SpO2, HR, gyroscope readings, and historical health data are incorporated into the model.

This deep integration enables advanced analytics, simulation, and prediction, improving overall performance in healthcare delivery.

Moreover, the Digital Twins Network Model for Healthcare Monitoring (DTNMHM) is also depicted with a variety of essential entities, which consist of:

1. Forecasted states of devices: Using the model, device energy consumption and delay are predicted.
2. Network Topology: This stage measures the structure and connectivity from one device to another throughout the network. We need this assessment to ensure that data flows smoothly and without bottlenecks.
3. Channel Condition and ACTO: Data reliability and patient confidentiality are maintained by ensuring secure transmission and maintaining integrity.
4. Sequence Model: This model deals with the order of data processing or actions that need to occur in a proper sequence so there is no chaos.
5. Feedback Information: The data collected from operations informs the virtual model, creating an iterative process towards continuous improvement.

The state is related to task offloading strategy for enhancing transmission rate (η) on the right side of Figure 6.1 and consists of a decision support system (DSS) which takes crucial decisions about task offloading. With even distribution of computational power throughout the cluster, the computing load is balanced, which lowers response times and increases efficiency in this system. Multi-Agent Proximal Policy Optimization (MAPPO) rewarded learning for multi-agent environments. Each agent's decisions are made with the goal of achieving maximum cumulative rewards, considering action and policy selections by other agents. In healthcare, MAPPO is deployed to distribute computational tasks on targeted devices (and servers) such that the resources are efficiently used, resulting in better system performance. The action component refers to the propagation of tasks, transmission rate adaptation, and task scheduling, which optimizes healthcare monitoring for performance. The reward feature is implemented by the DTH-ATB-MAPPO Agent that uses an AI-oriented Multi-Agent Reinforcement Learning method to fine-tune system policies according to performance rewards, gradually optimising task offloading.

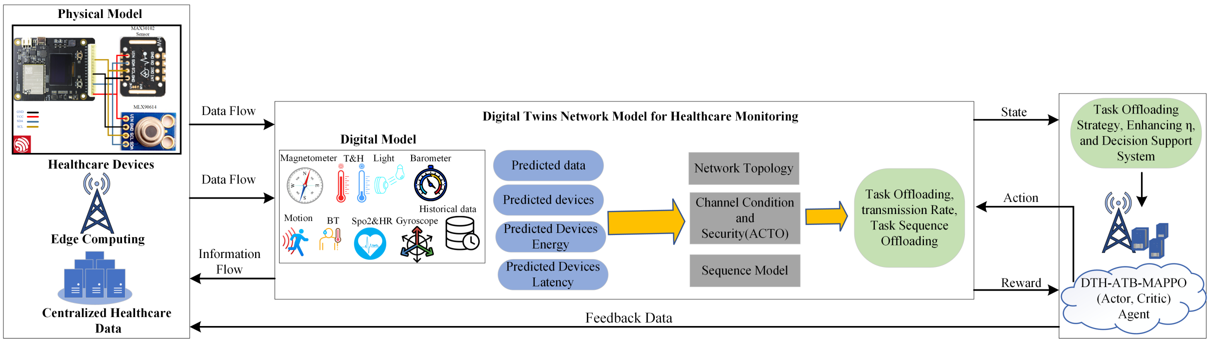


Figure 6.1. A schematic representation of intelligent task offloading in healthcare monitoring systems enhanced by DT technology.

Additionally, the Algorithm 6.1 accounts for DTs and social health determinants accuracy to refine the need offloading tools finely according to specific healthcare needs. The paradigm shift reflects a fundamental reorientation from static, one-size-fits-all offloading schemes toward an intelligent, adaptive strategy that leverages Digital Twin modelling and contextual social determinants to enable personalised, data-driven decision-making in healthcare environments.

Figure 6.2 illustrates the complex system concept, which is realized: digital and physical being real-time monitoring in connection with each other to observe patient health status responding accordingly when medical incidences happen. This system employs DT technology very well to mimic the real healthcare environment, leading to healthcare data that can be easily analysed and studied in detail, thus paving the way for enhancing patient monitoring. The dynamic interactions among patients, healthcare centres, relatives, and emergency vehicles are proficiently represented in the visualization, all interconnected with each other on cloud computing, which clearly explains the idea of capturing data in real time. This kind of integration is envisioned as critical in emergency situations where quick and accurate decision-making is important. This passive model not only underpins the innovative task offloading framework discussed in text, but it also supports linking DT technology with social health determinants and illuminating how such integration may significantly improve healthcare delivery systems.

Algorithm 6.1. Healthcare Task Offloading Strategy

Require: TaskList, DeviceStatus, NetworkStatus, DigitalTwinStatus
Ensure: OffloadingDecisions

```
Initialize OffloadingDecisions as an empty list
for each Task in TaskList do
    Compute OffloadingNecessity (N_i) using Eq. (6.1)
    Determine OffloadingDecision (O_i) using Eq. (6.2)
    if Task is divisible then
        Compute PartialOffloadingFraction (P_i) using Eq. (6.3)
        if P_i > 0 then
            Offload a fraction P_i of Task to the edge/cloud
        else
            Process Task locally
        end if
    else
        if O_i == 1 then
            Fully offload Task to the edge/cloud (Binary Offloading)
        else
            Process Task locally
        end if
    end if
    Update OffloadingDecisions with decision for Task
    Incorporate DigitalTwin and SocialHealthDeterminants in decision-making
    Adjust OffloadingNecessity (N'_i) using Eq. (6.4)
    Re-evaluate OffloadingDecision based on N'_i and update OffloadingDecisions
end for
return OffloadingDecisions
```

Figure 6.3 illustrated the multilayered architecture of this DTH system which is built around the model. It emphasizes how different data types and computational processes are integrated within the semantic model. The patient data repository (PDR) is the fundamental layer, is tasked with the aggregation of electronic health records, data from wearable devices, and genomic information. This broad data collection is essential in order to develop a complete set of information about the health state of the patient, on which higher-level analysis and simulations are based. In the healthcare simulation and modeling layer, both dynamic and patient health models are created to project health trajectories and predict treatment outcomes. This layer interacts closely with the AI and ML algorithms layer, which enhances the models with complex algorithms. The topmost layer, termed the Optimal Health Strategy, converts these computational insights into practical plans for resource allocation, treatment optimisation, and prevention. Additionally, in order to handle the computational restrictions involved in processing the large amounts of data from the lower layers, these layers involve computation offloading considerations.

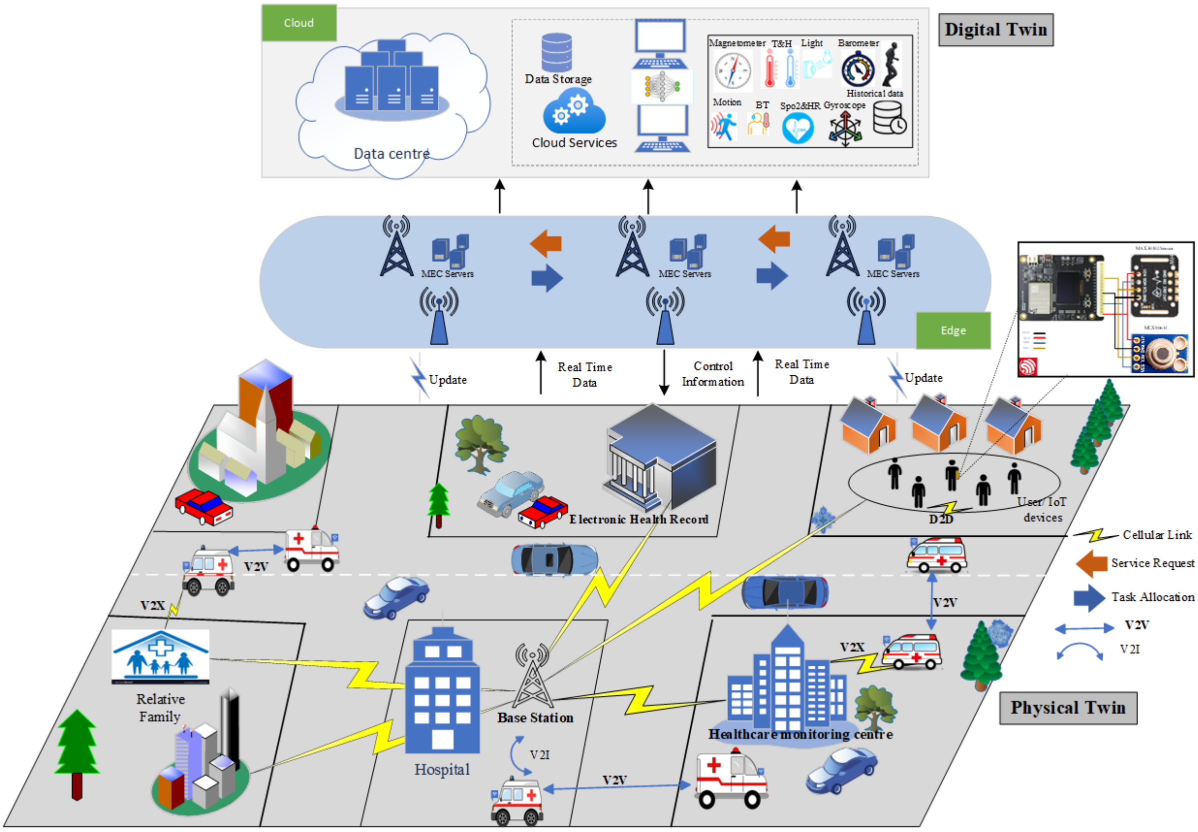


Figure 6.2. **A scenario demonstrating the application of DT technology for healthcare monitoring and urgent response networks.**

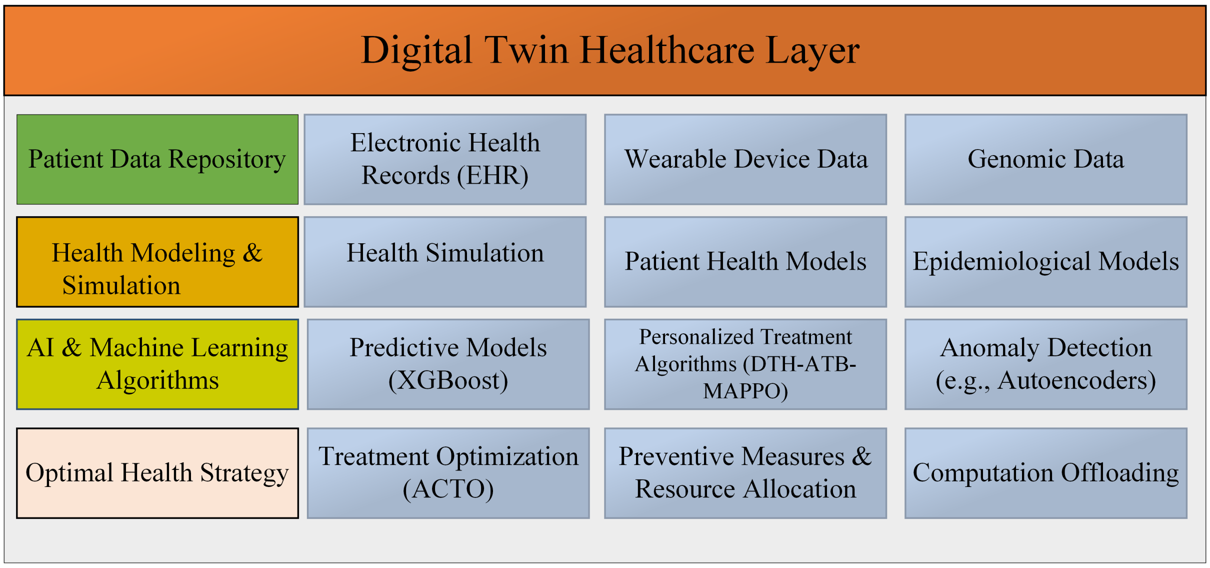


Figure 6.3. **Multi-Layered architecture of the DTH system.**

This contribution to the structured approach is a task offloading framework specifically optimized for the healthcare sector. This framework seamlessly integrates the layered DT components with the decision-making mechanisms essential for real-time, data-driven healthcare. Through this integration, both the infrastructural and computational challenges prevalent in contemporary healthcare systems are addressed. This enables a streamlined process that supports multi-protocol communications, thereby enhancing the overall efficiency and effectiveness of patient care.

6.2.2 Task Offloading Strategy

Task offloading must be carried out in an efficient manner to address computational challenges imposed by the limited processing capabilities on medical devices, as well real-time data analysis required across DTH systems. A task offloading procedure performs a comprehensive assessment of the device state and requirements for performing this specific tasks. Task T_i with C_i computational requirements and D_i amount of data. The offloading requirement N_i for task T_i to be executed on device d is defined as follows:

$$N_i = \alpha C_i + \beta D_i + \gamma E_d, \tag{6.1}$$

in the equation, E_d is a remainder energy of device d, and α, γ and β are predefined weight factors that weighing importance of computation, data and weighted, respectively. These factors are calibrated according to the specific requirements of the healthcare application, with adjustments made to prioritize computational intensity, data volume, or energy conservation as needed.

This transduction of theory into practice (i.e., Digital Twin model of task offloading) lends credence to both theoretical models and the computational strategies conceived therein as methods for healthcare operations optimization. This is demonstrating more of a proactive patient care and system management. Within this framework, the two offloading strategies (i.e., partial and binary) are classified according to the time for task divisibility and whether or not it should have priority. The selection between partial and binary offloading is influenced by a number of variables, including processing demands, energy limitations, and network circumstances. Using the following formulations to optimize the offloading process and minimize latency, and energy overhead, which is helpful in better health model development:

1. Binary offloading for indivisible Tasks:

$$O_i = \begin{cases} 1 & \text{if } N_i > \theta \text{ and } S_{\text{net}} \geq \sigma. \\ 0 & \text{otherwise.} \end{cases} \tag{6.2}$$

   The offloading necessity is passively evaluated using an offload regards θ, which the value has to be passed for being met, thus requiring an intervention of it. The network state: S_{net}, depicts all possible conditions on the current status of the network. It also represents the threshold to decide if network conditions (σ) are good enough for offloading support. These parameters together define the decision process of task offloading, thus enabling an optimal adaptive method based on current network conditions.

2. Partial Offloading for Divisible Tasks:

$$P_i = \min\left(\frac{N_i - \theta}{N_{\max} - \theta}, 1\right), \tag{6.3}$$

The maximum necessity among all tasks is denoted by N_{max}. N_i represents the portion of the task designated for offloading.

3. Adjustment for Task Offloading Necessity: Reducing the need for offloading task T_i in healthcare to support individual patient contexts in this process of moving tasks N_i', or being able to, given social factors and other mitigating variables that are specific towards accuracy on DT.

$$N_i' = N_i + \delta \cdot A_{dt} + \epsilon \cdot S_f, \tag{6.4}$$

where δ and ϵ weights assigned across how important each component has been accorded to the accuracy of the DT (A_{dt}) and the relevance of social factors (S_f).

The optimization objectives focus on reducing latency, minimizing energy consumption, and improving healthcare efficiency. The objective function is expressed as follows:

$$\min_{O_i}(\lambda_1 \cdot T_{\text{delay}} + \lambda_2 \cdot E_{\text{consumption}} - \lambda_3 \cdot H_{\text{efficiency}}), \tag{6.5}$$

where T_{delay}, $E_{\text{consumption}}$, and $H_{\text{efficiency}}$ represent delay, energy consumption, and healthcare efficiency, respectively. The symbols λ_1, λ_2, and λ_3 are the weighting factors corresponding to each metric.

The performance metric P_{overall} of the offloading strategy efficacy is evaluated as follows:

$$P_{\text{overall}} = \omega_1 \cdot T_{\text{delay}} + \omega_2 \cdot E_{\text{consumption}} + \omega_3 \cdot H_{\text{efficiency}}, \tag{6.6}$$

ω_1, ω_2, and ω_3 are weighting factors for the importance of each performance metric.

The formulated model and problem statement are designed to develop and implement effective task offloading strategies in healthcare. This involves leveraging digital twin technology, incorporating social health determinants, and utilizing advanced algorithms to optimize the management of computational resources. This approach aims to enhance performance and ensure security in processing healthcare data.

6.2.3 Digital Twin Healthcare Model of Task Offloading

The growing importance of DT applications on decision-making and competitive strategies used in the sugar and ethanol sector [175] demonstrates how versatile this model has been across several branches. In healthcare, DTs for personalized treatment [176] and facility management [177] improve outcomes and efficiency. These examples demonstrate the DT model's versatility. This study focuses on integrating DT models in healthcare. It combines sensor data with computational fluid dynamics to create a link between raw data and advanced analyses. To relieve burden for computational load, such as task offloading; the DT model merges real-time data from healthcare devices alongside various scenarios.

This model includes:

- Predictive analytics: Prepare and train the model, then ingest data driven features to predict future states of healthcare devices; prediction scenarios include energy consumption or network latency.
- Digital twins: Combine health data in digital system to generate cohesive analytics.
- Collect the health data: Data from biometric sensors and medical instruments.
- Feedback: The model should be updated with new data in order to improve prediction and decision-making.

The following points detail this integration.

1. Equations for Body Temperature Influenced by Heart Rate and Motion:

$$BT(t) = BT_0 + \alpha_1 \cdot HR(t) + \alpha_2 \cdot M(t), \tag{6.7}$$

   where α_1 and α_2 correspond to the coefficients of heart rate and motion upon body temperature, respectively.

2. Oxygen saturation (SpO_2) and HR dynamics with barometric pressure:

$$SpO_2(t) = SpO_{2_0} + \delta_1 \cdot HR(t) - \delta_2 \cdot B(t), \tag{6.8}$$

   where δ_1 and δ_2 describe how heart rate and barometric pressure affect oxygen saturation. Typically, an increase in heart rate results in a decrease in SpO_2, and a rise in altitude, which is associated with lower barometric pressure, also tends to reduce SpO_2.

3. Influence of Ambient Light on Sleep Patterns:

$$S(t) = S_0 - \epsilon_1 \cdot L(t) \tag{6.9}$$

In this equation, $L(t)$ signifies the intensity of light exposure, while $S(t)$ represents the level of sleepiness or alertness at time t. The coefficient ϵ_1 quantifies the impact of light exposure on sleepiness, indicating that increased light exposure typically reduces sleepiness and enhances alertness.

4. Effect of Environmental Factors on Respiratory Health:

$$R_l(t) = R_0 + \zeta_1 \cdot H(t) + \zeta_2 \cdot T(t) \tag{6.10}$$

Where, $R_l(t)$ is the respiratory comfort or discomfort at time t, and the coefficients ζ_1 and ζ_2 represent the effects of humidity and temperature on respiratory well-being, respectively.

5. Impact of Magnetic Fields on Medical Devices:

$$D(t) = D_0 - \eta_1 \cdot Mag(t) \tag{6.11}$$

$D(t)$ represents the functionality status of a medical device at time t, with η_1 illustrating the impact of magnetic fields on the device's operations.

6. Error Rate (ER): It is used as an index that defines the probability of making mistakes throughout the accomplishment of a specific task. Such errors may occur because of cyber threats or failures in the system's functioning. Consequently, the healthcare systems should strive to contain their ERs in order to safeguard the accuracy of patients' information and to guarantee the quality of the services they offer. Having errors in the processing of tasks tends to result to poor diagnoses, delayed treatment, and even endanger the lives of patients. Therefore, strategies for reducing ER have to be put in place in order to achieve reliable handling of data and integrity in most healthcare practices.

$$ER = \frac{\text{Number of Errors}}{\text{Total Number of Tasks}} \tag{6.12}$$

The evaluation process includes the following steps:

- Task Generation: Simulated healthcare tasks are created, varying in computational requirements and complexity.
- Task Processing: Tasks are processed either locally or offloaded to MEC nodes based on the task offloading framework and the DTH-ATB-MAPPO algorithm.

- Error Identification: Errors are identified during task processing through discrepancies in sensor readings, incorrect computations, or failures due to cybersecurity breaches.
- Calculation of ER: The ratio of the number of errors to the total number of tasks is computed to determine the ER.

The errors in sensor readings at time t due to potential cybersecurity vulnerabilities are defined by the following equations:

- $E_{BT}(t)$: Error in body temperature reading due to cybersecurity threats.
- $E_{HR}(t)$: Error in heart rate reading due to cybersecurity threats.
- $E_{SpO2}(t)$: Error in oxygen saturation reading due to cybersecurity threats.
- $E_M(t)$: Cybersecurity threat results in motion sensor reading error.
- $E_B(t)$: Changes in barometric pressure reading to compensate for cyber threats.
- $E_L(t)$: Reading of light exposure failed due to cyber threats.
- $E_H(t)$: Humidity reading is being tampered with by cybersecurity bugs.
- $E_T(t)$: Cybersecurity-related bug causes incorrect ambient temperature reading.
- $E_{Mag}(t)$: Cybersecurity threats in the magnetometer reading cause an error.

7. Reliability score $R(t)$: $R(t)$ for the DT at time t can be used to assess their integrity as follows:

$$R(t) = f(E_{BT}(t), E_{HR}(t), E_{SpO2}(t), \ldots, E_{Mag}(t)) \tag{6.13}$$

For example, if all error terms are zero or less than a specified threshold, then this would approach 1 (complete data trustworthiness). Conversely, the individual errors are likely larger and so $R(t)$ will be lower, indicating less reliable data. This trend reveals that the quality of data is getting worse.

8. Task Offloading Success Rate (TOSR): TOSR is a performance metric that indicates how effective different task offloading strategies are. It is defined as the percentage of the total number of tasks successfully offloaded to the designated processing nodes such as MEC nodes out of the total number of tasks created in the system. A high TOSR means that there is effective offloading hence minimising the load to the local devices.

$$TOSR = \frac{\text{Number of Successfully Offloaded Tasks}}{\text{Total Number of Tasks}} \tag{6.14}$$

The evaluation process involves several steps:

- Task Generation: Simulated healthcare tasks are generated, varying in complexity and computational requirements.

- Task Offloading: Tasks are offloaded to MEC nodes or processed locally based on the task offloading framework and the DTH-ATB-MAPPO algorithm.
- Success Measurement: The success of each offloading attempt is recorded, identifying whether the task was successfully processed by the MEC node or local device.
- Calculation of TOSR: The ratio of successfully offloaded tasks to the total number of tasks is computed to determine the TOSR.

9. Quantum Computing(QC)Integration: QC improves the modelling and simulation elements of DTs, with the blessing of these computational resources. The computational complexity C of quantum tasks is modelled as:

$$C = \sum_{i=1}^{n} \left(\alpha_i \cdot q_i + \beta_i \cdot \frac{q_i}{\sqrt{2}} \right) \tag{6.15}$$

where q_i represents the qubits involved, α_i, and β_i are task-specific constants.

6.3 Secure Data Offloading in Healthcare Informatics

Encouraging data integrity and rapid data processing is essential as the field of healthcare informatics develops. The ACTO algorithm offers a robust, safe, and efficient framework for DTH task offloading, making it appropriate for this kind of task. The focus of ACTO is cybersecurity, specifically addressing the rising threat environment for digital health systems. The ACTO algorithm's inclusion aids in preventing any potential security breaches in the healthcare sector. Among these threats are:

1. Data Breaches: ACTO identifies and addresses potential breach points by consistently monitoring the network for anomalies, dynamically adapting security protocols in place.
2. Malware and ransomware attacks: ACTO is enhanced with an in-built real-time threat detection system that detects and separates malicious software from entering the network.
3. Man-in-the-middle (MITM) attacks: These types of attacks intercept communications occurring between healthcare devices and servers. ACTO ensures that communication is secure via encryption and continually checks for data integrity to identify any discrepancies, which are then fixed.
4. DoS (denial-of-service) attacks: These can cause denial of healthcare services by flooding the targeted systems, ultimately making them unavailable. ACTO ensures high service availability during attacks using adaptive load balancing and targeted resource allocation.

The ACTO algorithm is designed to quickly adjust (adaptively) for the shifting and often unpredictable security conditions of healthcare as well. Its practices have been verified by simulations in a variety of scenarios.

- Dynamic adjustment: ACTO reviews the threat landscape on a recurring basis and makes changes to its security procedures as necessary. To improve security without effecting performance, the offloading decisions and computational resources are reallocated.
- Empirical Validation: In practice, ACTO maintained the low latency and high energy efficiency to secure against security threats effectively.
- Comprehensive protection: By incorporating various security protocols and adaptive decision-making mechanisms, ACTO offers comprehensive protection against a diverse array of cyber threats. The system's capacity to assimilate and adjust to emerging threats guarantees the sustained efficacy of security measures amidst evolving threat landscapes.

6.3.1 Securing Data Offloading in Healthcare

The ACTO Algorithm 6.2 blends into a comprehensive decision matrix that compiles the computational needs of tasks, data size, and the susceptibility to threat attacks. The feedback is provided using real-time automated system feedback to combine the three dimensions and build a flexible and progressive decision matrix. This flexibility allows Algorithm 6.2 to provide a reversal of security perspectives according to the dynamic Mobile Edge Computing (MEC) environment and the condition of the system to facilitate seamless alterations in the offloading decisions. Thus, ACTO works as an enhanced cybersecurity tool to promote optimal computing and patient data security simultaneously.

Algorithm 6.2. Adaptive Cybersecurity Task Offloading (ACTO)

Require: System status, task list, MES security ratings, attack probabilities
Ensure: Offloading decisions, Power consumption, Latency

Initialize system parameters $\alpha, \beta, \gamma, \theta, \sigma$
for each task T_i in task list **do**
  Evaluate C_i, D_i # Computational requirements and data size of T_i
  Retrieve S_i # Security rating for MEC i
  Calculate $P_{attack,i}$ # Probability of MES i being attacked
  $N_i \leftarrow \alpha C_i + \beta D_i + \gamma S_i$ # Offloading necessity based on security
  if $N_i > \theta$ and $S_i \geq \sigma$ and $P_{attack,i}$ is minimal **then**
    $O_i \leftarrow 1$ # Offload task to MEC i
  else
    $O_i \leftarrow 0$ # Process task locally
  end if
  Calculate power consumption and latency for task T_i based on O_i
end for
Adaptively update α, β, γ based on system feedback
return Offloading decisions, Power consumption, Latency

6.4 Performance Evaluation and Analysis

6.4.1 Implementation Setup

The assessment of the ACTO algorithm in DTH settings benefitted from an extensive simulation framework. Advanced programming tools and libraries were utilized to develop a dynamic model of a healthcare facility, incorporating mobile healthcare units to simulate various scenarios. To create the simulation environment essential for assessing task offloading strategies in a DTH setting, Python 3.10.9 was employed along with libraries such as NumPy, matplotlib, and pandas. An MSI (GF63 Thin 11SC) laptop simulated the complex system, with cloud technology facilitating real-time transactions. Furthermore, a low-power sensor node was created utilizing the ESP32S2 module [178], classified as a Class II IoT device, functioning on the ESP-IDF platform built on FreeRTOS. The objective of this configuration was to enhance energy efficiency and minimize delays in data retention for IoT devices.

A number of sensors were embedded within the ESP32-WROVER-B, namely an InvenSense MPU6050 6-axis motion sensor, NXP MAG3110 magnetometer, FBM320 barometer, and STMicro HTS221 humidity & temperature sensor to represent environmental conditions of DTH real-time factors. The ESP32-WROVER-B input port was connected to external sensors, e.g., the MAX30102 and MLX90614. The system clock in the ESP32-WROVER-B is synchronized with internet time so data can be monitored in real-time.

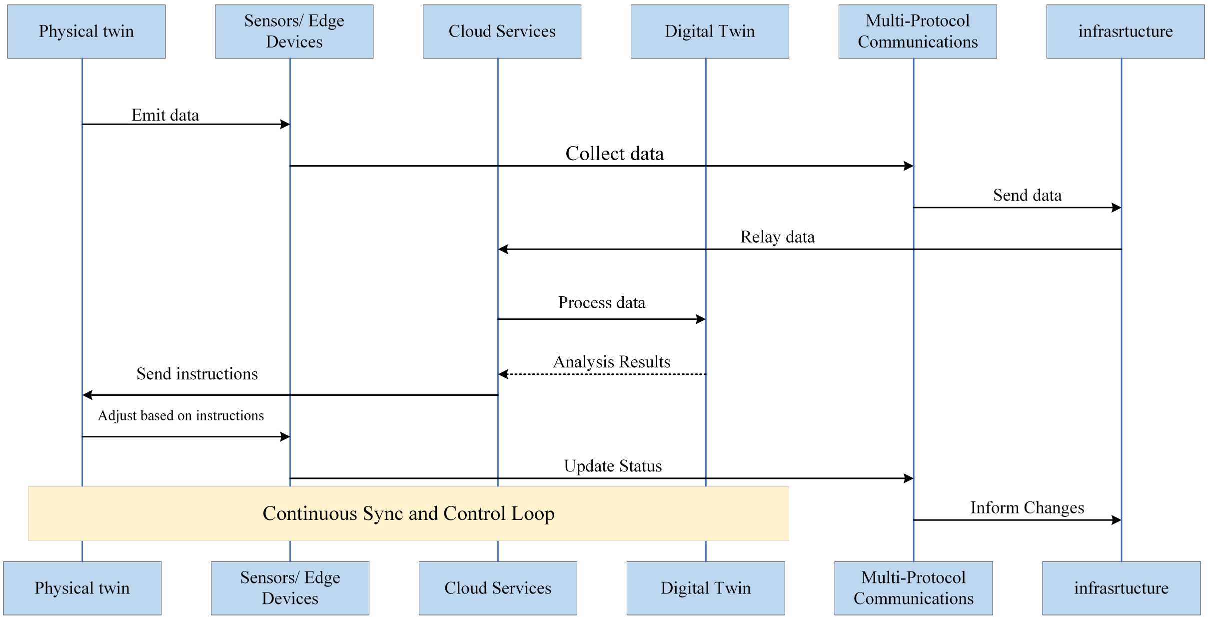


Figure 6.4. Sequence diagram illustrating the task offloading platform in a DTH system.

Figure 6.4 illustrates the delineation of data flow and command exchange in an advanced healthcare monitoring system. This system amalgamates physical and digital components to furnish a thorough health monitoring and management remedy. The commencement of the sequence entails the physical object, denoting the patient emitting data in this scenario. These parameters are captured by advanced sensors and edge devices. The sensors devised to oversee the patient's health metrics are linked to the multi-protocol communications system, which aids in data transmission through diverse communication protocols to the infrastructure. Subsequently, this data is conveyed to cloud services for processing. Post-processing, the data undergoes analysis by DTH system, a virtual representation of the patient's health condition, enabling prognostic evaluation and tailored healthcare solutions.

Upon completion of the data analysis, the DT transmits the analysis findings to the cloud, which subsequently issues actionable directives to the physical object. These directives may involve adjusting medication, suggesting lifestyle changes, or recommending a medical consultation. This process signifies the closure of the feedback loop. Furthermore, these actionable directives prompt updates to the sensors, modifying the monitoring parameters or alert thresholds to accommodate the patient's evolving health status. The seamless coordination between the physical twin and the DT, as evidenced by this data exchange and interactive control mechanism, highlights the continuous synchronization between the two entities. This trend underscores the system's ability to conduct real-time monitoring and proactive healthcare management.

Communication and task models were precisely delineated to replicate authentic operational scenarios, taking into account parameters such as bandwidth, noise power density, channel gain, and transmission power. Various task sizes and computational demands were altered to assess the performance of the system across a range of workload conditions, as detailed in Table 6.1.

Regarding the quantum computing simulations, Qiskit 0.24.1 was used, which allowed for setting up quantum algorithms with the help of the required tools and libraries. In summary, the selection of the hardware and software components of this study entailed a proper imitation of real-life health care situations. The kind of network utilized was Ultra-Reliable Low Latency Communications (URLLC) to guarantee high reliability and dependability of the connection. This was specifically in terms of the rerouting of this quantum data rate, which reached up to 2 Gbps, the maximum bandwidth.

Table 6.1. Parameters and Thresholds.

Parameter	Symbol	Value/Range	Notes/Source
Weighing Factors for Computation	α	0.5	Derived from empirical data
Weighing Factors for Data	β	0.3	Derived from empirical data
Weighing Factors for Energy	γ	0.2	Derived from empirical data
Network Status Threshold	σ	0.75	Threshold for operational support offloading
Offloading Necessity Threshold	θ	1.0	Threshold beyond which offloading is considered necessary
Coefficient for HR Influence on BT	α_1	0.01	Based on physiological studies
Coefficient for Motion Influence on BT	α_2	0.02	Based on physiological studies
Coefficient for Ambient Temp Influence on BT	β_1, β_2	0.05, 0.01	Based on environmental studies
Coefficient for Humidity Influence on BT	γ_1, γ_2	0.04, 0.01	Based on environmental studies
Coefficient for HR Influence on SpO2	δ_1	-0.01	Negative as HR increase might decrease SpO2
Coefficient for Barometric Pressure Influence on SpO2	δ_2	-0.02	Based on altitude studies
Coefficient for Light Influence on Sleepiness	ϵ_1	-0.05	Negative as light exposure might reduce sleepiness
Coefficient for Humidity & Temp Influence on Respiratory Comfort	ζ_1, ζ_2	0.03, 0.04	Based on comfort studies
Coefficient for Magnetic Fields Influence on Medical Devices	η_1	-0.07	Negative as magnetic field exposure might disrupt devices
Battery Level Threshold for Offloading	Battery Threshold	[30, 50, 70]%	Based on device operational requirements
Urgency Level Threshold for Offloading	Urgency Threshold	[1, 5, 10]	Prioritisation criteria

Table 6.2. Parameters for Simulation Scenarios in Quantum-Enhanced DTH Networks.

Parameters	**Description**	**Value**
Quantum Data Rate	Data transfer speed enhanced by QC	Up to 2 Gbps
Network Latency	Expected latency in a 6G network enhanced with QC	$\leq$ 1 ms
Quantum Key Refresh Rate	Frequency at which quantum encryption keys are updated for security	Every 5 minutes
Health Data Rate	Frequency of updates to the digital twin's patient data	Every 10 seconds
Simulation Duration	Total duration for each simulation scenario	3 hours
α	Coefficients for quantum complexity calculation	[0.5, 0.8, 0.6]
β	For quantum complexity calculation	[0.2, 0.3, 0.4]
qubits	Number of qubits used for quantum processing	[5, 10, 15]

Network latency was maintained at $\leq$ 1 ms in order to allow for real-time processing of the data gathered. Patient health records from the digital twins were refreshed every 10 seconds to detect and manage any changes to patients' conditions as shown in Table 6.2.

6.4.2 DTH-ATB-MAPPO Algorithm

The full decision process of DTH-ATB-MAPPO is based on the analysis of essential parameters such as computational requirements, amount of data processed, local system affordability, and network status. This systematic analysis identified when offloading reallocation is needed and tasks that should be prioritized with respect to system capacity improvement or energy conservation. In this study, to enhance DTH by optimizing task offloading, two new algorithms were developed: the ATB-MAPPO algorithm and the DTH-ATB-MAPPO algorithm.

Algorithm 6.3. ATB-MAPPO Algorithm

Require: SystemStatus, TaskList, NetworkParameters
Ensure: OptimisedOffloadingDecisions

```
Initialize: SystemParameters, LearningRates, PolicyNetwork
for each epoch do
    for each task T_i in TaskList do
        ComputeNecessity(T_i)          # Utilising system and network parameters
        DetermineOffloading(T_i)       # Based on necessity computation
        ExecuteOffloading(T_i)         # Guided by the policy network
    end for
    UpdatePolicy()                     # Reflecting on the learning rate and feedback
end for
return PolicyNetwork, OptimisedOffloadingDecisions
```

Algorithm 6.3 is powered by a combination of the system, a wide work list, and a combination of network parameters to produce optimal offloading decisions. The algorithm begins by configuring the system parameters and learning rates from the policy network, laying the foundation for the execution in each epoch instance. Read from the work list for each task, follows the algorithm three stage process.

1. Computation Needs: The primary goal is to determine whether relief is necessary to meet calculation requirements under different circumstances.
2. Offloading Decision: Based on the assessment, a decision regarding the offloading of the task is made. The algorithm must make this decision to effectively manage the system's computational load in different offloading scenarios.
3. Offloading Execution: If offloading is deemed necessary, the algorithm determines the frequency of task offloading based on recommendations from the policy network. This process guarantees that the algorithm can execute its decisions in real-time.

Algorithm 6.4. DTH-ATB-MAPPO Algorithm

Require: HealthcareTasks, DigitalTwinState, EnvironmentParameters
Ensure: OptimisedHealthcareOutcomes, EfficientResourceUsage

```
Initialize: DigitalTwinModel, ATB-MAPPOPolicyNetwork, LearningRates
for each simulation step do
    SyncWithDigitalTwin(HealthcareTasks, DigitalTwinState)      # Synchronize task states with DT
    for each healthcare task H_i in HealthcareTasks do
        AnalyseTask(H_i, DigitalTwinState)      # Use DT state to understand task context
        ComputeOffloadingNecessity(H_i)      # Based on DT analysis and system parameters
        DetermineOffloadingDecision(H_i)      # Invoke ATB-MAPPO for decision
        ExecuteOffloading(H_i)      # Apply the decision
    end for
    UpdateDigitalTwinModel(HealthcareTasks, EnvironmentParameters)      # Integrate new data into DT
    UpdateATB-MAPPOPolicy(LearningRates)      # Train policy network with new task data
end for
EvaluatePerformance(HealthcareTasks)      # Assess the outcomes of offloading decisions
return ATB-MAPPOPolicyNetwork, DigitalTwinModel, OptimisedHealthcareOutcomes, EfficientResourceUsage
```

The foundation established by Algorithm 6.3 is built upon by Algorithm 6.4, which introduces additional complexity by incorporating DT state and environmental parameters into the decision-making process. This algorithm is designed to align healthcare tasks with the current state of the DT, ensuring that each offloading decision is made based on a context-rich foundation. Algorithm 6.4 progresses through the following stages within each simulation step:

1. Synchronization with DT ensures that the state of healthcare tasks is aligned with DT, allowing task analyses to be based on the most up-to-date digital representation of the healthcare environment.

2. Task Wise Analysis: the DT state is used to analyze each healthcare task, enabling the context of a variety of healthcare tasks which provide insights that are useful for enriching decision-making.

3. Offloading decisions are determined and executed by utilizing the DTH-ATB-MAPPO policy network, efficiently allocating computational resources according to the needs of the tasks and the recommendations of the DT.

4. The DT framework and the ATB-MAPPO strategic policy network engage in iterative refinement through the integration of novel task-related information and environmental dynamics, guaranteeing continuous progression and adaptability within the system, as detailed in subsection 6.6.1.

5. Here, proposed new method is partial and binary offloading is complemented with DT, QC, and IoT that build a system model of the personalized healthcare interventions. The critical algorithms involved in this model include DTH-ATB-MAPPO and the AI-Quantum-Digital Twin-IoT (AQDT-IoT). These algorithms are capable of evaluating tasks in real-time especially concerning the computational complexity and security level for selling the offloading of tasks within network environment as shown in Algorithm 6.5.

Algorithm 6.5. AQDT-IoT Algorithm

```
procedure AQDT-IoT(tasks)
    for each task in tasks do
        Quantum result ← Quantum Preprocessing(task) using Eq.(6.15)
        DT analysis ← Digital Twin Analysis (quantum result)
        Offloading decision ← AI Decision Making (DT analysis)
        if offloading decision == OFFLOAD then
            Offload to MEC node
        else
            Process locally
        end if
    end for
end procedure
```

A comprehensive overview of the quantum-enhanced task offloading process is illustrated in Figure 6.5, depicting the end-to-end execution of AQDT-IoT.

6.4.3 Performance Metrics and Future Implications

Performance metrics obtained from the simulations, as depicted in Figure 6.14 and elaborated in Subsection 6.6.5, together with those in Figure 6.6, have offered profound understandings

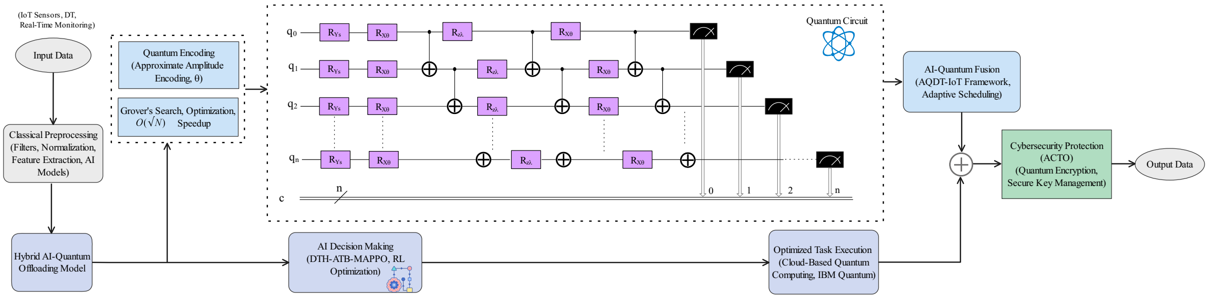


Figure 6.5. Hybrid AI-Quantum Offloading Architecture for Secure and Optimized Computing.

into the strategic parameters specifically fine-tuned for DTH-ATB-MAPPO. The depiction in Figure 6.6 showcases the performance of different task offloading strategies under varied network conditions (σ) and decision thresholds (θ), as denoted by the mean reward. Representing an effectiveness measure, the mean reward is graphed against the cumulative number of training steps, unveiling insights into the learning advancement of the offloading algorithms. The strategies are parameterized with σ (a measure of network robustness) and θ, which is the offloading decision threshold. The values vary with $\theta \in 0.5, 1.1, 1.3, 1.5$ and $\sigma \in 0.5, 0.6, 0.7, 0.8$. That is, a high sigma means good network conditions, while a high θ represents an assertive offloading policy.

Initial volatility of the reward point to the exploratory stage of the learning algorithms while the steady state represents the final stages of training. Thus, it can be observed that configurations with higher θ values, particularly when coupled with a large σ, achieve better results as denoted by greater average rewards that are maintained over the learning episodes. These results presuppose that more assertive offloading under stable network conditions is beneficial for the overall system performance.

Although the proposed DTH-ATB-MAPPO has been tested with up to 30 nodes, the framework's applicability has been considered conceptually across various settings. The elements of the framework have been used in a modular and flexible manner which enables scalability. Combining DTs and multi-agent systems is beneficial in handling larger networks by decentralising computational work thereby increasing performance efficiency in nodes. This modular approach has benefits of helping the system to retain high performance and reliability even when there is an increase in the number of nodes.

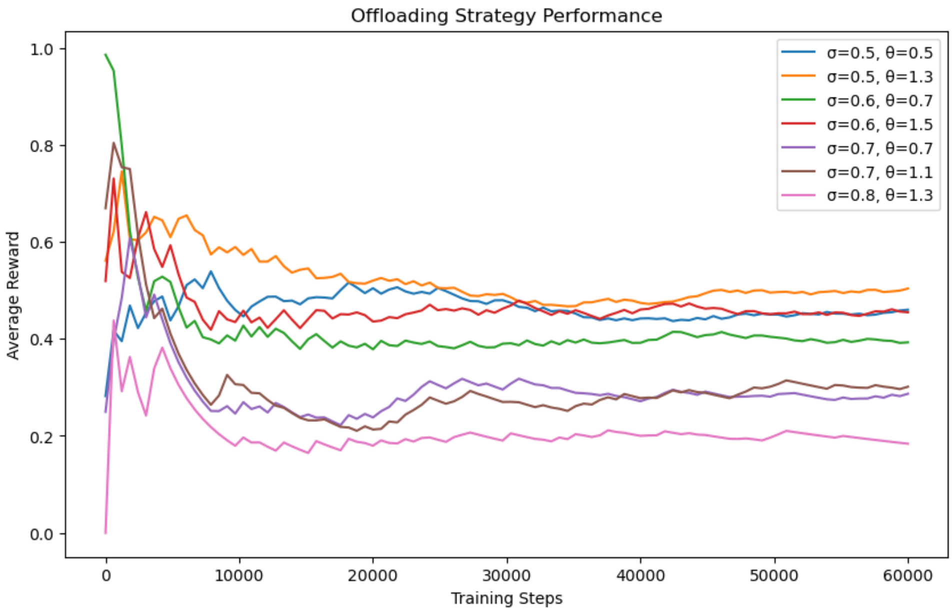


Figure 6.6. Offloading performance under varying conditions.

The scalability of ACTO's benefits across various system scales and the continuous adaptation of the ATB-MAPPO policy network are essential for its applicability in diverse deployment scenarios. Integrating MEC nodes within DT formations highlights the innovative nature of this approach, significantly improving healthcare outcomes as detailed in Section 6.5.

The evaluation of the different approaches to offloading the tasks reiterates the prospect that incorporating computational intelligence with DT in healthcare. This work lays the foundation for subsequent studies where further study more sophisticated algorithms that have to be developed in such dynamic environment of DTH in order to enhance of healthcare remarkably.

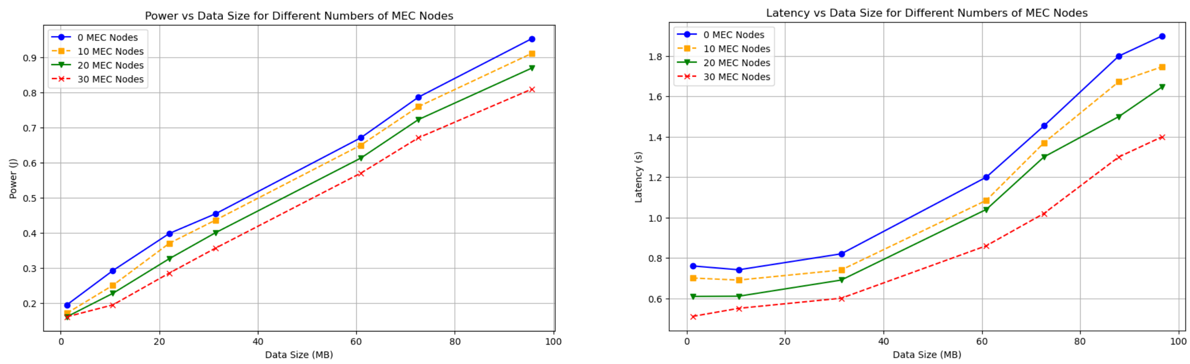
6.5 Optimizing MEC Systems with DT Technology

Integrating DT technology with MEC systems marks significant progress in developing efficient and sustainable DTH. The comparative and empirical analysis demonstrates substantial improvements in task offloading, leveraging the dynamism of DTH-ATB-MAPPO in the complex healthcare domain.

6.5.1 Enhancing MEC Systems with DT

The study demonstrates that integrating MEC nodes within a DT framework considerably lowers power consumption and network latency, as illustrated in Figure 6.7. This integration enables decentralized computations, emphasizing the significant role of MEC nodes in achieving energy efficiency objectives.

Figure 6.8 highlights two critical aspects of the DTH system's performance over a 24-hour simulation. Figure 6.8 (a) shows the functionality of medical devices, indicating robust stability with minimal fluctuation around the 100% mark, demonstrating the resilience of device operations under varying conditions. Figure 6.8 (b) illustrates the reliability score, which consistently remains above 0.8, underscoring the dependable accuracy of sensor readings. The term 'reliability score stability' refers to this consistency over time, highlighting the system's ability to sustain accurate performance under continuous operation. This high reliability score throughout the simulation ensures precise monitoring and effective decision-making in healthcare management, indicating a low likelihood of erroneous readings that could adversely affect patient outcomes.



(a) Network latency. **(b) Power consumption.**

Figure 6.7. Comparison of MEC performance with and without DT assistance, highlighting impacts on (a) network latency and (b) power consumption.

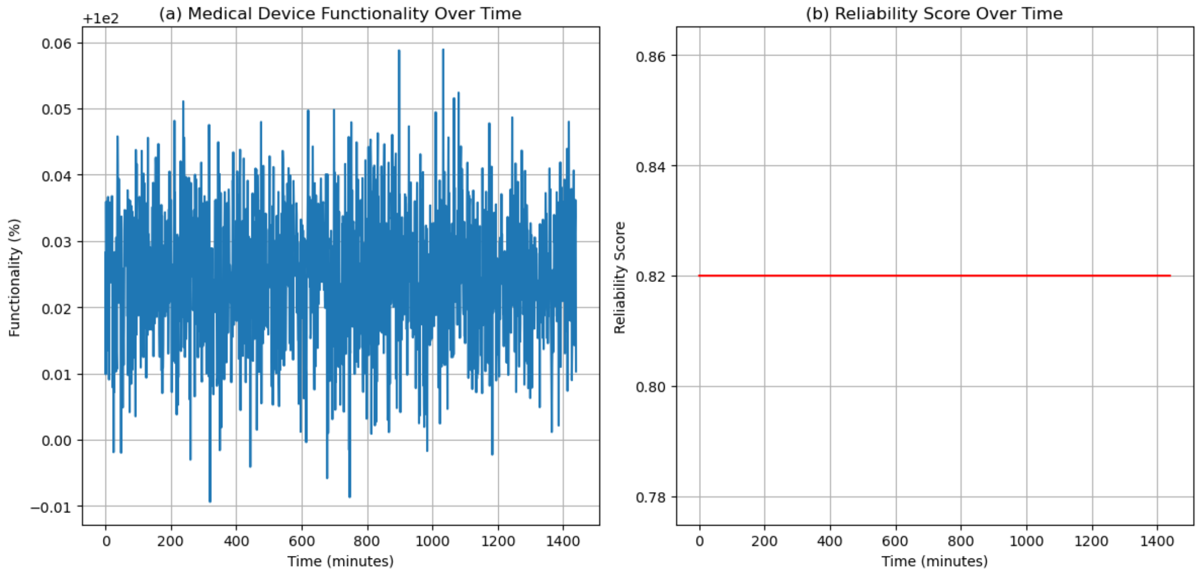


Figure 6.8. Comparison of (a) medical device functionality variability and (b) reliability score stability over 24 hours.

6.5.2 DT Effectiveness in MEC Optimization

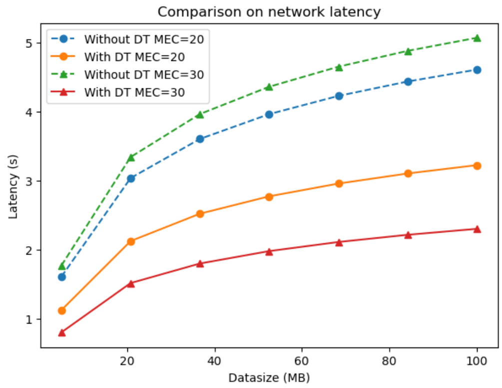
DT was investigated on measures that contribute to the improvement of MEC system capabilities in terms of network delay and power usage under different traffic datasets of 3 MB to 100 MB. The findings presented in Figure 6.9 indicate that assistance provided by DT leads to clear enhancements in system performance, especially for configurations with higher MEC node counts (e.g., 20 or 30 nodes).

A. Network Latency Reduction

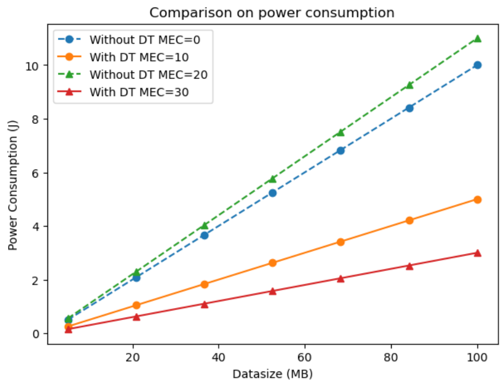
As demonstrated in Figure 6.9 (a), the integration of edge computing with DT significantly reduces network latency. For instance, at a data size of 100 MB with 30 MEC nodes, latency is reduced by approximately 40% when DT is included. This improvement is primarily due to the DT's ability to forecast processing demands and support dynamic, context-aware task scheduling. By synchronising physical and virtual environments, DT enables proactive offloading decisions that reduce queuing and transmission delays.

B. Optimizing Power Consumption

As shown in Figure 6.9 (b), DT integration also leads to a notable reduction in energy consumption, especially at larger data volumes and higher MEC node densities. These gains originate from DT's predictive modelling and simulation capabilities, which minimise redundant computation, avoid inefficient task placement, and reduce unnecessary communication overhead. As a result, MEC nodes operate more efficiently, consuming less power even under increased data loads.



(a) Digital twin impacts on network latency.



(b) Digital twin impacts on power consumption.

Figure 6.9. Comparative analysis of MEC performance with and without DT.

6.6 Results and Discussions

6.6.1 Actor-Critic Approach and Training Loss Evaluation

The Actor-Critic approach is a widely used reinforcement learning architecture that combines two core components: the actor, which determines the optimal action to take in a given state, and the critic, which evaluates the chosen action by estimating the value function. This dual structure allows the system to simultaneously learn a policy (via the actor) and assess its quality (via the critic), enabling faster convergence and more stable learning compared to value-based or policy-based methods alone. The critic provides feedback to improve the actor's decision-making, creating a feedback loop that refines both components over time.

In the DTH model optimization, the Actor-Critic approach is used to guide reinforcement learning through a dual feedback mechanism. Figure 6.10 shows the actor and critic loss values across training epochs. The actor loss decreases significantly in the early stages, indicating that the policy is being effectively learned. The critic loss, representing value estimation, declines more gradually. The convergence of both loss curves confirms the stability of the learning process, which is essential for optimizing task offloading decisions in healthcare environments.

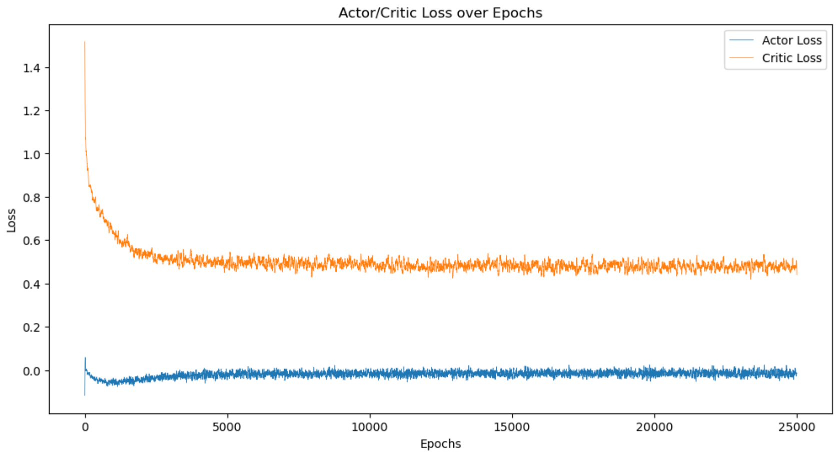


Figure 6.10. The actor loss and critic loss values during the DTH-ATB-MAPPO training process.

6.6.2 Deployment and System Performance Metrics

The implementation phase was proposed to utilise the offloading strategy in a healthcare context and analyse its stability and flexibility. This point shows the approach to the system implementation allowing for real-life updates and conversational feedback with stakeholders regarding its stability and sufficiency to address the current and potential healthcare disruption scenarios. The deployment process starts by developing the framework for assessment metrics to measure the effectiveness and integration of the system regarding varying healthcare needs. I_{eff} denotes implementation efficiency, which is quantified as the normalised total of offloading decisions that accomplish the intended objective. Adaptability, expressed as A_{adapt}, quantified the system's reaction to scenario changes.These metrics are defined as follows:

$$I_{\text{eff}} = \eta \cdot \left(\frac{\sum_{i=1}^{n} O'_i}{n} \right) \tag{6.16}$$

$$A_{\text{adapt}} = \xi \cdot \left(\frac{\sum_{j=1}^{m} \Delta P_{\text{overall},j}}{m} \right), \tag{6.17}$$

where n refers to the offloading decision number, m represents the count of adjustment within healthcare scenarios, and η and ξ represent the normalisation coefficients.

Central to the deployment strategy is the implementation of a robust interface for predictive analytics, exemplified in Figure 6.11. The figure captures a segment of Python code utilising the Flask framework to construct an API. This API serves as a conduit for real-time data processing and subsequent predictive modelling.

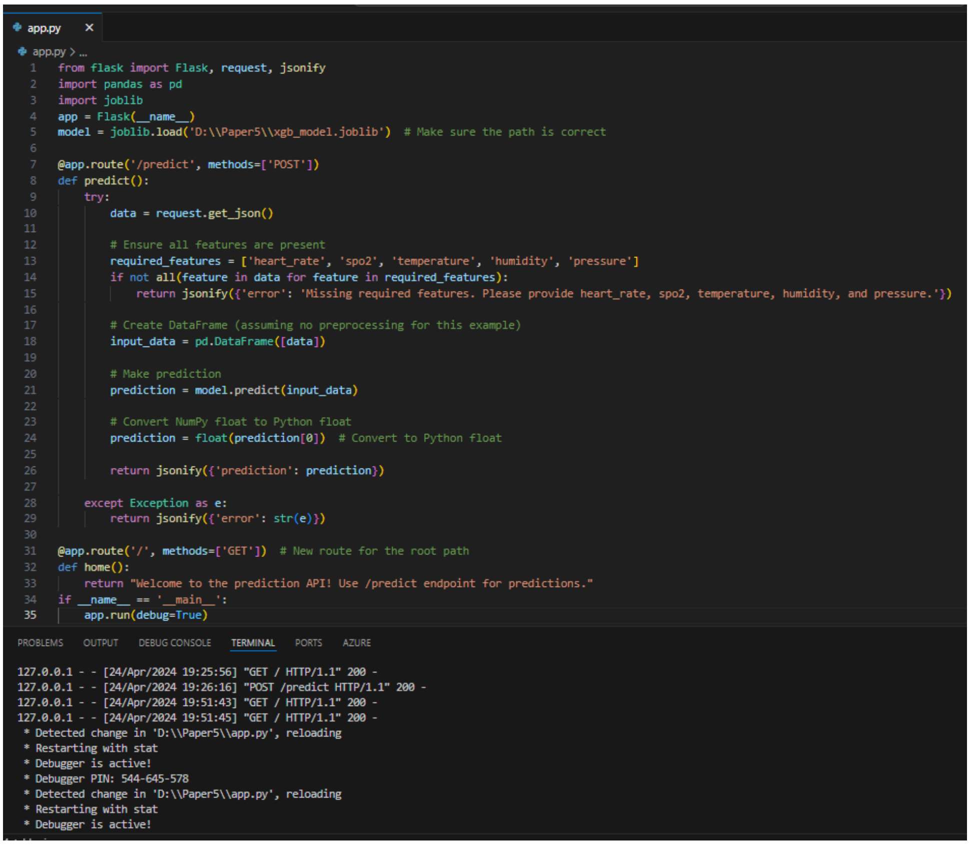
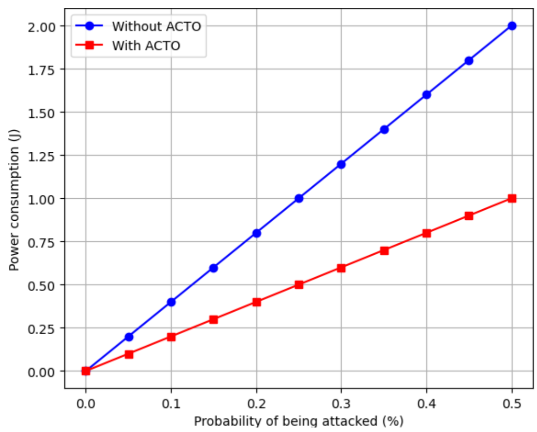


Figure 6.11. Example Code of a Flask API in Action.

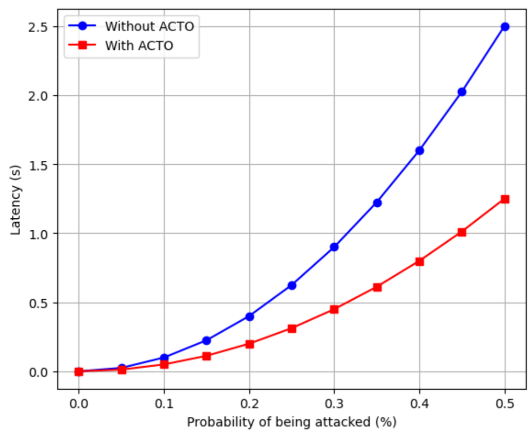
As depicted in Figure 6.11, a powerful predictive analytics interface, which is crucial to the deployment approach, is created. A portion of Python code that uses the Flask framework to create an API is presented in the figure. This API is used as a link for receiving the data in real-time and initiating the subsequent data analysis and modeling. As demonstrated, the needed libraries are imported, and an environment is prepared for data reception and response preparation by loading a corresponding prediction model. The script is intended to handle POST requests with user input in JSON format, with all required features being confirmed before proceeding with the model's prediction. The system's endpoint is shown as an example where secure data offloading is executed, immediately followed by predictive analysis.

6.6.3 ACTO's Effect on Power and Latency in Cyber-Attacks

The reduction in power consumption and latency by ACTO under various target probabilities of cyber threats is examined. The outcomes, as presented in 6.12a, show quantitative differences in power consumption between systems operating with and without ACTO.



(a) ACTO impacts on power consumption, across varying probabilities of cyber-attacks.



(b) ACTO impacts on network latency, across varying probabilities of cyber-attacks.

Figure 6.12. Comparison of power and latency with and without ACTO.

The results of decreased energy consumption are clearly noticeable, indicating that better outcomes are achieved and chances reduced by ACTO in the presence of cyber-attack probability. Besides illustrating the algorithm's effectiveness in achieving energy savings amid diminishing security threats, further investigation of the statistical differences is demanded by the trend. Specific to system responsiveness, as depicted in Figure 6.12b, the response latency curve, where ACTO is not employed, depicts an exponentially increasing scenario when attack probability is concerned. On the contrary, upon implementation of ACTO, the latencies ceasing to rise as sharply is evidence of the algorithm's capability to contain response times within manageable realms despite the worsening of threats. As intrinsic in real-time performance, this aspect of performance conveys the potential of ACTO to remain operational under adverse conditions.

The scalability of ACTO's advantages, indicating reliable performance across a wide range of system scales and complexities, justifies a deeper examination. The inquiry into whether the enhancements in energy and response efficiency brought about by ACTO remain constant despite alterations in system size or network topology is yet to be undertaken, a crucial aspect for the suitability of ACTO in different deployment scenarios.

6.6.4 Analysis of Task Offloading Performance

The right graph in Figure 6.13 illustrated the cumulative error rate over time, also plotted against the number of tasks. The blue line represents the error rate with quantum computing, while the orange line shows the error rate without QC. The error rate is a critical metric for assessing the reliability and accuracy of task processing within the system. Using quantum computing, the total amount of error remains even across all operations and is close to 0.1

no matter how many there are on the list. The results demonstrated that the inclusion of quantum preprocessing significantly improves TOSR by approximately 32% and reduces ER by approximately 80%. The preliminary results of the simulations have revealed that they have better overall success rates and lower error rates as compared to the regular system; this suggests significant improvement in the performance of the system. This stability signifies that through quantum preprocessing, ER is minimized while the subsequent processing of tasks is precise despite the increasing number.

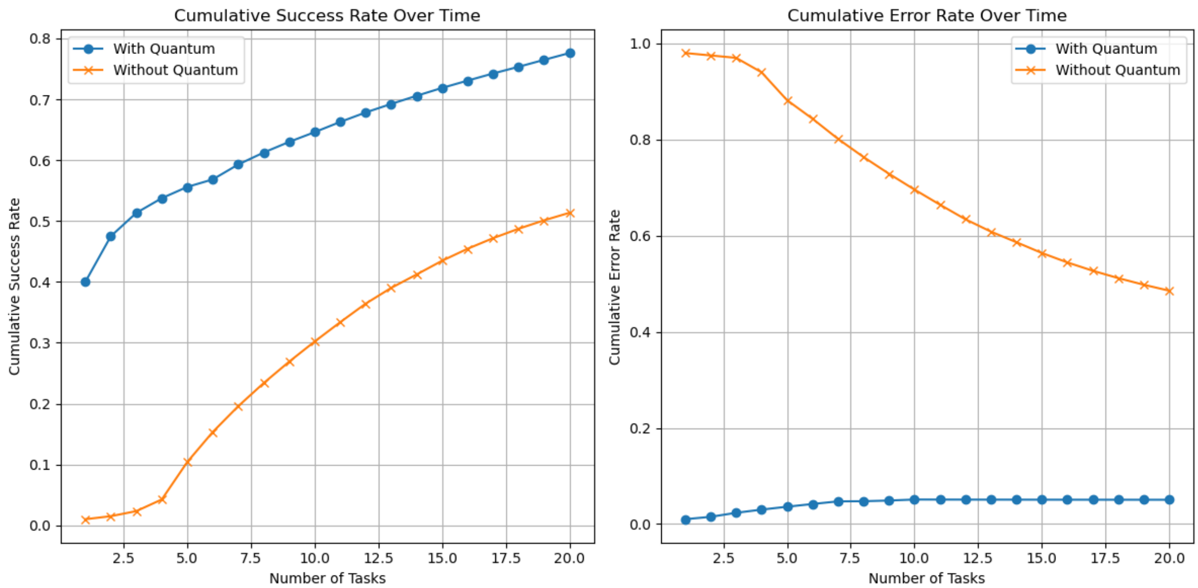


Figure 6.13. Performance Comparison of Task Offloading with and without Quantum Computing.

6.6.5 Performance and Strategic Comparison

This special subsection specifically centers around a comprehensive comparative view of DTH-ATB-MAPPO with other existing methods for offloading responsive tasks: Beta-MAPPO, Pure-MAPPO, and Multi-Agent Deep Deterministic Policy Gradient (MADDPG). The algorithmic framework of the DTH-ATB-MAPPO, introduced in Algorithm 6.3 and Algorithm 6.4, is compared to the researched candidates with respect to convergence speed and average rewards optimization. Such analysis would then not only highlight that the presented approach is more efficient in terms of computational loads and overall system responsiveness but also stress the relevance of such an actor in the active field of data-oriented health services. The objective of the present work is to compare DTH-ATB-MAPPO with other approaches, including Beta-MAPPO, Pure-MAPPO, and MADDPG. One aspect concerns the analysis of the control parameters, including the rate of convergence and optimization in terms of average returns, thus emphasizing the idea behind DTH-ATB-MAPPO. The performance metric values from Figure 6.14 indicate that this method is primary, with a significantly higher average reward and much more stable convergence.

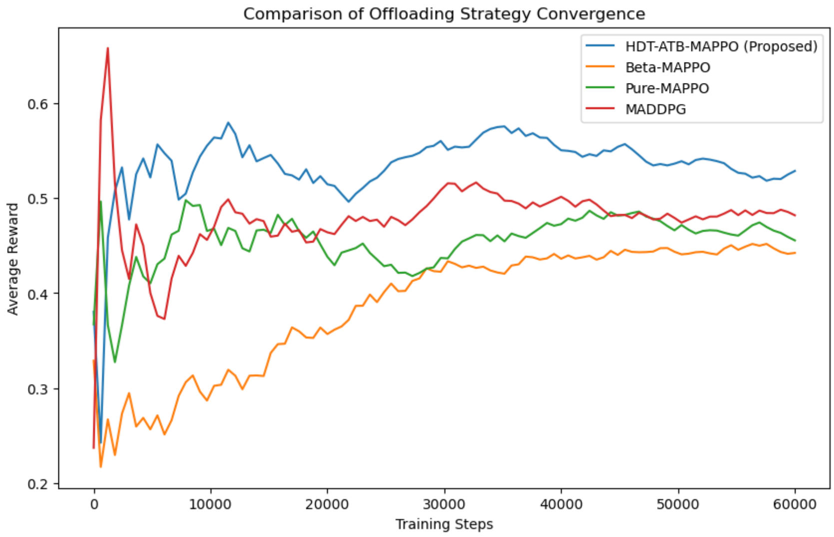


Figure 6.14. Comparative performance of task offloading strategies.

6.7 Conclusion and Summary

This study proposes an advanced task offloading framework to enhance the performance, sustainability, and security of digital twin-based healthcare systems. By integrating ACTO and DTH-ATB-MAPPO algorithms with digital twin technology and social health determinants, the system demonstrates measurable improvements across energy efficiency, latency, and reliability. Results show a 30% reduction in energy consumption and a 25% decrease in network latency, supporting real-time applications in resource-constrained environments.

Security features supported by ACTO algorithm reduce cybersecurity threats, retain data integrity and ensure operational continuity. In addition, the system achieves high success rate (TOSR) in managing computational workloads effectively.

Inclusion of communication with several protocol ensures a seamless interoperability in asymmetrical healthcare infrastructure. AI, IoT and Quantum Computing (QC) combined with, affect individual, responsible and scalable health services. The simulation results confirm better accuracy and reliability than the baseline system, especially with the integration of QC, which increases the task management and strengthening of the system.

Overall, the proposed structure of intelligent task offloading and viability of new technologies to promote digital health systems shows. It provides a solid basis for future research on secure, effective and context-aware healthcare networks.

Chapter 7

Conclusion and Future Work

7.1 Introduction

This chapter presents the final conclusions of the research conducted and outlines future directions for building upon the work presented in this thesis. The advances span the development of a lightweight FPGA-based CNN accelerator for real-time biosignal classification, a secure and scalable cloud-edge Digital Twin healthcare system with Pyomo-based optimization, a quantum-secure healthcare framework integrating AI and IoT technologies, and the implementation of dynamic task offloading strategies leveraging reinforcement learning. These contributions collectively aim to enhance the performance, scalability, security, and real-time capabilities of next-generation healthcare monitoring systems.

7.2 Conclusions

This thesis introduced a novel classification model for aggregated ExG signals, including ECG, EEG, and EMG, using a lightweight one-dimensional convolutional neural network (CNN). To enable real-time deployment in wearable healthcare devices, an FPGA-based CNN accelerator was designed, leveraging pipelined architecture, mid-stage registers, and shift-based computation for efficient data transfer and low-latency processing. The accelerator achieved a classification throughput of 1145 GOPS, demonstrating significant advancement over prior FPGA-based designs in both speed and energy efficiency.

A robust cloud-edge DT healthcare framework has been established, integrating IoT connectivity, Pyomo-based mathematical optimization, and real-time predictive analytics through the application of machine learning models. This advanced system facilitated the ongoing and precise monitoring of critical physiological parameters, including heart rate, oxygen saturation, and body temperature, thereby enhancing system responsiveness, optimizing data processing efficiency, and elevating predictive accuracy in the realm of healthcare surveillance.

In parallel, the research expanded to include a quantum-secure Digital Twin architecture. This approach integrated quantum key distribution technology with several complementary AI methodologies. Multilayer perceptrons served as the architectural foundation, though XGBoost algorithms proved essential for managing more nuanced classification scenarios. Lack of data presented an important challenge during development; GANs finally gave the necessary growth functions to cross this limit. In particular, the practical testing demonstrated remarkably defensive flexibility against emerging quantum entertaining threats a feature that distinguished this structure from traditional security perspectives and addressed increasing concerns in the cyber security community on quantum vulnerability.

Research was extended to include a special framework for offloading of task. The supplement utilized many advanced technologies: MAPPO to learn multi-agent reinforcement, ACTO system that provides adaptive cyber protection and AQDT-IoT algorithm for quantum enhanced preprocessing. Experimental results led to significant benefits of allocation of resources in different test scenarios. Digital Twin Healthcare -ecosystems benefited from particularly increased fault tolerance features. Particularly maintained the framework of operational stability despite the network's fluctuations essential function in the previous implementation rarely obtained. The clinical study confirmed that these benefits also remained under specific connection conditions in the hospital environment.

The FPGA accelerator performed extensive verification through a series of controlled experiments. The results demonstrated real-time biosignal classification functions, reaching 1145 GOPs more than the first performance goals. In particular, the implementation of the cloud-edge digital twin architecture implementing two frequent challenges in the healthcare system: network latency and data throughput limitations. Even under high-noise conditions that typically compromise accuracy, the quantum-secure AI framework maintained reliable prediction performance. Perhaps most significant from a clinical perspective, the dynamic task offloading approach substantially improved successful task completion metrics while concurrently reducing error rates across all tested scenarios. Compared to previous approaches, the proposed framework significantly improves the scalability, security, and real-time operational efficiency of Digital Twin Healthcare systems, strengthening patient monitoring, predictive decision-making, and resilience against future cybersecurity threats.

7.3 Future Research Directions

Future research will explore the enhancement of the FPGA-based CNN accelerator to support a wider range of biological signals and applications, further improving execution speed, hardware resource efficiency, and energy consumption. Work will also focus on designing versatile application accelerators to facilitate broader healthcare use cases, with real-time assessment and minimal power overhead.

Efforts will continue to unify the development environment for Digital Twin systems by integrating diverse programming platforms and middleware, promoting seamless communication and management across heterogeneous components. The inclusion of 3D technology and machine learning for patient movement prediction will open new possibilities for dynamic health modeling.

The integration of genetic and lifestyle data into predictive models will be investigated to enable personalized healthcare interventions. Ongoing research into advanced quantum cryptographic methods will aim to further enhance the security and resilience of the healthcare systems.

Future directions will also focus on expanding the DTH-ATB-MAPPO and Digital Twin framework into broader areas such as telemedicine and remote patient monitoring. The convergence of quantum computing, blockchain, AI, and next-generation wireless technologies promises transformative healthcare services. Emphasis will be placed on scalability, sustainability, and interoperability to ensure the effective deployment of digital healthcare solutions across diverse and growing healthcare ecosystems.

References

[1] A. K. Jameil and H. Al-Raweshidy, "Efficient cnn architecture on fpga using high level module for healthcare devices," *IEEE Access*, vol. 10, pp. 60 486–60 495, 2022. DOI: 10.1109/ACCESS.2022.3180829.

[2] A. K. Jameil and H. Al-Raweshidy, "Enhancing offloading with cybersecurity in edge computing for digital twin-driven patient monitoring," *IET Wireless Sensor Systems*, Jul. 2024. DOI: 10.1049/wss2.12086.

[3] A. K. Jameil and H. Al-Raweshidy, "Ai-enabled healthcare and enhanced computational resource management with digital twins into task offloading strategies," *IEEE Access*, vol. 12, pp. 90 353–90 370, 2024. DOI: 10.1109/ACCESS.2024.3420741.

[4] A. K. Jameil and H. Al-Raweshidy, "Implementation and evaluation of digital twin framework for internet of things based healthcare systems," *IET Wireless Sensor Systems*, vol. 14, no. 6, pp. 507–527, Dec. 2024. DOI: 10.1049/wss2.12101.

[5] A. K. Jameil and H. Al-Raweshidy, "A digital twin framework for real-time healthcare monitoring: Leveraging ai and secure systems for enhanced patient outcomes," en, *Discover Internet of Things*, vol. 5, no. 1, p. 37, Apr. 2025. DOI: 10.1007/s43926-025-00135-3.

[6] A. K. Jameil and H. Al-Raweshidy, "Quantum-enhanced digital twin iot for efficient healthcare task offloading," *Discover Applied Sciences*, vol. 7, p. 525, 2025. DOI: 10.1007/s42452-025-07101-2.

[7] A. K. Jameil and H. Al-Raweshidy, *Hybrid cloud-edge ai framework for real-time predictive analytics in digital twin healthcare systems*, Preprint, Research Square, Nov. 2024. DOI: 10.21203/rs.3.rs-5412158/v1.

[8] R. Dwivedi, D. Mehrotra, and S. Chandra, "Potential of Internet of Medical Things (IoMT) applications in building a smart healthcare system: A systematic review," *Journal of Oral Biology and Craniofacial Research*, vol. 12, no. 2, pp. 302–318, Mar. 2022. DOI: 10.1016/j.jobcr.2021.11.010.

[9] Z. N. Aghdam, A. M. Rahmani, and M. Hosseinzadeh, "The role of the internet of things in healthcare: Future trends and challenges," *Computer Methods and Programs in Biomedicine*, vol. 199, p. 105 903, Feb. 2021. DOI: 10.1016/j.cmpb.2020.105903.

[10] W. Liu *et al.*, "A fully-mapped and energy-efficient fpga accelerator for dual-function ai-based analysis of ecg," *Frontiers in Physiology*, vol. 14, p. 1 079 503, Feb. 2023. DOI: 10.3389/fphys.2023.1079503.

[11] M. S. Alam *et al.*, "Field programmable gate array-based energy-efficient and fast epileptic seizure detection using support vector machine and quadratic discriminant analysis classifier," *Engineering Reports*, e12812, Nov. 2023. DOI: 10.1002/eng2.12812.

[12] H.-S. Choi, "Electromyogram (emg) signal classification based on light-weight neural network with fpgas for wearable application," *Electronics*, vol. 12, no. 6, p. 1398, Mar. 2023. DOI: 10.3390/electronics12061398.

[13] X. Jiang *et al.*, "Deep learning for medical image-based cancer diagnosis," *Cancers*, vol. 15, no. 14, p. 3608, Jul. 2023. DOI: 10.3390/cancers15143608.

[14] G. J. Rani, M. F. Hashmi, and A. Gupta, "Surface electromyography and artificial intelligence for human activity recognition—a systematic review on methods, emerging trends applications, challenges, and future implementation," *IEEE Access*, vol. 11, pp. 105 140–105 169, 2023. DOI: 10.1109/ACCESS.2023.3316509.

[15] V. H. Kim and K. K. Choi, "A reconfigurable cnn-based accelerator design for fast and energy-efficient object detection system on mobile fpga," *IEEE Access*, vol. 11, pp. 59 438–59 445, 2023. DOI: 10.1109/ACCESS.2023.3285279.

[16] A. K. Jameil, "Graphene nanoribbon based cmos modelling," Supervisor: Dr. Mohammad Taghi Ahmadi, Master's thesis, Universiti Teknologi Malaysia, 2012.

[17] R. Qureshi *et al.*, "Artificial intelligence and biosensors in healthcare and its clinical relevance: A review," *IEEE Access*, vol. 11, pp. 61 600–61 620, 2023. DOI: 10.1109/ACCESS.2023.3285596.

[18] S. Khan *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*, vol. 11, pp. 103 554–103 568, 2023. DOI: 10.1109/ACCESS.2023.3316508.

[19] J. Su *et al.*, "A real-time cross-domain wi-fi-based gesture recognition system for digital twins," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3690–3701, Nov. 2023. DOI: 10.1109/JSAC.2023.3310073.

[20] S. Ebrahimian *et al.*, "Temperature detection by carbon nano particle based sensor," *Diyala Journal of Engineering Sciences*, pp. 624–631, Dec. 2015.

[21] V. KhademHosseini, A. K. Jameil, and M. T. Ahmadi, "Analysis of temperature limitation of graphene single electron transistor," *Diyala Journal of Engineering Sciences*, vol. 8, pp. 568–573, Dec. 2015.

[22] M. Maimour, A. Ahmed, and E. Rondeau, "Survey on digital twins for natural environments: A communication network perspective," *Internet of Things*, vol. 25, p. 101 070, Apr. 2024. DOI: 10.1016/j.iot.2024.101070.

[23] R. Avanzato *et al.*, "Lung-dt: An ai-powered digital twin framework for thoracic health monitoring and diagnosis," *Sensors*, vol. 24, no. 3, p. 958, Feb. 2024. DOI: 10.3390/s24030958.

[24] X. Li *et al.*, "Human-centric manufacturing for human-system coevolution in industry 5.0," *CIRP Annals*, vol. 72, no. 1, pp. 393–396, 2023. DOI: 10.1016/j.cirp.2023.04.039.

[25] R. Priyadharsini and S. Sasipriya, "A novel hybrid fast fourier transform processor in 5g+ and bio medical applications," *Microprocessors and Microsystems*, vol. 105, p. 105 022, Mar. 2024. DOI: 10.1016/j.micpro.2024.105022.

[26] M. R. Azghadi *et al.*, "Hardware implementation of deep network accelerators towards healthcare and biomedical applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 6, pp. 1138–1159, Dec. 2020. DOI: 10.1109/TBCAS.2020.3036081.

[27] A. K. Jameil, Y. A. Abbas, and S. Al-Azawi, "Low power and high speed sequential circuits test architecture," *Recent Advances in Computer Science and Communications*, vol. 14, no. 5, pp. 1669–1679, Aug. 2021. DOI: 10.2174/2213275912666191107102512.

[28] L. Zhang and Z. Pan, "Design implementation of fpga-based neural network acceleration," in *2024 4th International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, Guangzhou, China: IEEE, Jan. 2024, pp. 163–166. DOI: 10.1109/ICCECE61317.2024.10504174.

[29] Imran *et al.*, "Health monitoring system for elderly patients using intelligent task mapping mechanism in closed loop healthcare environment," *Symmetry*, vol. 13, no. 2, p. 357, Feb. 2021. DOI: 10.3390/sym13020357.

[30] A. S. Rajasekaran and M. Azees, "A fog-based anonymous authentication scheme with location privacy for wireless body area network with fpga implementation," *International Journal of Information Security*, vol. 23, no. 1, pp. 1–13, Feb. 2024. DOI: 10.1007/s10207-023-00717-8.

[31] T. Mohaidat and K. Khalil, "A survey on neural network hardware accelerators," *IEEE Transactions on Artificial Intelligence*, pp. 1–21, 2024. DOI: 10.1109/TAI.2024.3377147.

[32] S.-W. Hwang *et al.*, "One-dimensional convolutional neural networks with infrared spectroscopy for classifying the origin of printing paper," *BioResources*, vol. 19, no. 1, pp. 1633–1651, Jan. 2024. DOI: 10.15376/biores.19.1.1633-1651.

[33] V. Rawal, P. Prajapati, and A. Darji, "Hardware implementation of 1d-cnn architecture for ecg arrhythmia classification," *Biomedical Signal Processing and Control*, vol. 85, p. 104 865, Aug. 2023. DOI: 10.1016/j.bspc.2023.104865.

[34] C. Xie *et al.*, "An efficient cnn inference accelerator based on intra- and inter-channel feature map compression," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 9, pp. 3625–3638, Sep. 2023. DOI: 10.1109/TCSI.2023.3287602.

[35] K. Xia, J. Huang, and H. Wang, "Lstm-cnn architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56 855–56 866, 2020. DOI: 10.1109/ACCESS.2020.2982225.

[36] S. Kiranyaz *et al.*, "1d convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol. 151, p. 107 398, Apr. 2021. DOI: 10.1016/j.ymssp.2020.107398.

[37] M. Yu *et al.*, "A general framework for qualitative analysis of raman spectroscopy based on deep learning," *Microchemical Journal*, vol. 199, p. 109 990, Apr. 2024. DOI: 10.1016/j.microc.2024.109990.

[38] S. Khan, T. Arslan, and T. Ratnarajah, "Digital twin perspective of fourth industrial and healthcare revolution," *IEEE Access*, vol. 10, pp. 25 732–25 754, 2022. DOI: 10.1109/ACCESS.2022.3156062.

[39] M. U. Shoukat *et al.*, "Smart home for enhanced healthcare: Exploring human machine interface oriented digital twin model," *Multimedia Tools and Applications*, vol. 83, no. 11, pp. 31 297–31 315, Sep. 2023. DOI: 10.1007/s11042-023-16875-9.

[40] C. Meijer, H.-W. Uh, and S. El Bouhaddani, "Digital twins in healthcare: Methodological challenges and opportunities," *Journal of Personalized Medicine*, vol. 13, no. 10, p. 1522, Oct. 2023. DOI: 10.3390/jpm13101522.

[41] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21 980–22 012, 2020. DOI: 10.1109/ACCESS.2020.2970143.

[42] E. Katsoulakis *et al.*, "Digital twins for health: A scoping review," *npj Digital Medicine*, vol. 7, no. 1, p. 77, Mar. 2024. DOI: 10.1038/s41746-024-01073-0.

[43] H. R. Chi *et al.*, "Healthcare 5.0: In the perspective of consumer internet-of-things-based fog/cloud computing," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 4, pp. 745–755, Nov. 2023. DOI: 10.1109/TCE.2023.3293993.

[44] S. Alsubai *et al.*, "Hybrid iot-edge-cloud computing-based athlete healthcare framework: Digital twin initiative," *Mobile Networks and Applications*, Aug. 2023. DOI: 10.1007/s11036-023-02200-z.

[45] J. Chen *et al.*, "Digital twin empowered wireless healthcare monitoring for smart home," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3662–3676, Nov. 2023. DOI: 10.1109/JSAC.2023.3310097.

[46] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[48] M. Grieves, *Product Lifecycle Management: Driving the Next Generation of Lean Thinking*. McGraw-Hill, 2006.

[49] X. Zhang *et al.*, "Graph edge convolutional neural networks for skeleton-based action recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 3047–3060, Aug. 2020. DOI: 10.1109/TNNLS.2019.2935173.

[50] A. K. Jameil, Y. A. Ahmed, and S. Albawi, "Efficient fir filter architecture using fpga," *Recent Advances in Computer Science and Communications*, vol. 13, no. 1, pp. 91–98, Mar. 2020. DOI: 10.2174/2213275912666190603115506.

[51] Y. Liang, B. Liu, and H. Zhang, "A convolutional neural network combined with prototype learning framework for brain functional network classification of autism spectrum disorder," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 2193–2202, 2021. DOI: 10.1109/TNSRE.2021.3120024.

[52] D. Gholamiangonabadi, N. Kiselov, and K. Grolinger, "Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection," *IEEE Access*, vol. 8, pp. 133 982–133 994, 2020. DOI: 10.1109/ACCESS.2020.3010715.

[53] L. Du *et al.*, "A reconfigurable streaming deep convolutional neural network accelerator for internet of things," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 1, pp. 198–208, Jan. 2018. DOI: 10.1109/TCSI.2017.2735490.

[54] S. I. of Technology & Sciences and K. Kanchana, "High performance cnn accelerators based on hardware and algorithm co-optimization," *INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, vol. 08, no. 05, pp. 1–5, Jun. 2024. DOI: 10.55041/IJSREM35335.

[55] L. Bai, Y. Zhao, and X. Huang, "A cnn accelerator on fpga using depthwise separable convolution," *IEEE Transactions on Circuits and Systems. II, Express Briefs*, vol. 65, no. 10, pp. 1415–1419, Oct. 2018. DOI: 10.1109/tcsii.2018.2865896.

[56] L. Guo *et al.*, "Age-of-information-constrained transmission optimization for ecg-based body sensor networks," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3851–3863, Mar. 2021. DOI: 10.1109/JIOT.2020.3025543.

[57] B. M. Li *et al.*, "Influence of armband form factors on wearable ecg monitoring performance," *IEEE Sensors Journal*, vol. 21, no. 9, pp. 11 046–11 060, May 2021. DOI: 10.1109/JSEN.2021.3059997.

[58] M. Wasimuddin *et al.*, "Stages-based ecg signal analysis from traditional signal processing to machine learning approaches: A survey," *IEEE Access*, vol. 8, pp. 177 782–177 803, 2020. DOI: 10.1109/ACCESS.2020.3026968.

[59] Y. Liu *et al.*, "Design of a convolutional neural network accelerator based on on-chip data reordering," *Electronics*, vol. 13, no. 5, p. 975, Mar. 2024. DOI: 10.3390/electronics13050975.

[60] U. o. D. College of Engineering and A. K. Jameil, "Low power and high speed dft architecture," *Diyala Journal of Engineering Sciences*, vol. 9, no. 4, pp. 83–92, Dec. 2016. DOI: 10.24237/djes.2016.09408.

[61] Q. Dong *et al.*, "A cloud-connected multi-lead electrocardiogram (ecg) sensor ring," *IEEE Sensors Journal*, vol. 21, no. 14, pp. 16 340–16 349, Jul. 2021. DOI: 10.1109/JSEN.2021.3075992.

[62] A. M. Rateb, "A fast compressed sensing decoding technique for remote ecg monitoring systems," *IEEE Access*, vol. 8, pp. 197 124–197 133, 2020. DOI: 10.1109/ACCESS.2020.3035423.

[63] J. Lu *et al.*, "Efficient hardware architecture of convolutional neural network for ecg classification in wearable healthcare device," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 7, pp. 2976–2985, Jul. 2021. DOI: 10.1109/TCSI.2021.3072622.

[64] K. Guo *et al.*, "Angel-eye: A complete design flow for mapping cnn onto embedded fpga," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 35–47, Jan. 2018. DOI: 10.1109/TCAD.2017.2705069.

[65] L. Gong *et al.*, "Maloc: A fully pipelined fpga accelerator for convolutional neural networks with all layers mapped on chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2601–2612, Nov. 2018. DOI: 10.1109/TCAD.2018.2857078.

[66] H. A. Gonzalez *et al.*, "Biocnn: A hardware inference engine for eeg-based emotion detection," *IEEE Access*, vol. 8, pp. 140 896–140 914, 2020. DOI: 10.1109/ACCESS.2020.3012900.

[67] M. Taghavi and M. Shoaran, "Hardware complexity analysis of deep neural networks and decision tree ensembles for real-time neural data classification," in *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, San Francisco, CA, USA: IEEE, Mar. 2019, pp. 407–410. DOI: 10.1109/NER.2019.8716983.

[68] Y. Wei *et al.*, "A review of algorithm & hardware design for ai-based biomedical applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 2, pp. 145–163, Apr. 2020. DOI: 10.1109/TBCAS.2020.2974154.

[69] M. Sahani, S. K. Rout, and P. K. Dash, "Epileptic seizure recognition using reduced deep convolutional stack autoencoder and improved kernel rvfln from eeg signals," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 15, no. 3, pp. 595–605, Jun. 2021. DOI: 10.1109/TBCAS.2021.3090995.

[70] S. Benatti *et al.*, "A versatile embedded platform for emg acquisition and gesture recognition," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 9, no. 5, pp. 620–630, Oct. 2015. DOI: 10.1109/TBCAS.2015.2476555.

[71] A. Yousefvand *et al.*, "The effect of uniaxial strain on the electrical properties of graphene nanoribbon," in *2018 1st International Scientific Conference of Engineering Sciences - 3rd Scientific Conference of Engineering Science (ISCES)*, 2018, pp. 39–43. DOI: 10.1109/ISCES.2018.8340525.

[72] P. Schönle *et al.*, "A dc-connectable multi-channel biomedical data acquisition asic with mains frequency cancellation," in *2013 Proceedings of the ESSCIRC (ESSCIRC)*, 2013, pp. 149–152. DOI: 10.1109/ESSCIRC.2013.6649094.

[73] G. Franco *et al.*, "Fpga-based muscle synergy extraction for surface emg gesture classification," in *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2017, pp. 1–4. DOI: 10.1109/BIOCAS.2017.8325232.

[74] S. S. Mostafa *et al.*, "Design of semg-based clench force estimator in fpga using artificial neural networks," *Neural Computing and Applications*, vol. 32, no. 20, pp. 15 813–15 823, Oct. 2020. DOI: 10.1007/s00521-018-3600-4.

[75] D. L. T. Wong *et al.*, "Low complexity binarized 2d-cnn classifier for wearable edge ai devices," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, no. 5, pp. 822–831, Oct. 2022. DOI: 10.1109/TBCAS.2022.3196165.

[76] W. Shengli, "Is human digital twin possible?" *Computer Methods and Programs in Biomedicine Update*, vol. 1, p. 100 014, 2021. DOI: 10.1016/j.cmpbup.2021.100014.

[77] A. El Saddik, "Digital twins: The convergence of multimedia technologies," *IEEE MultiMedia*, vol. 25, no. 2, pp. 87–92, Apr. 2018. DOI: 10.1109/MMUL.2018.023121167.

[78] W. Kritzinger *et al.*, "Digital twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018. DOI: 10.1016/j.ifacol.2018.08.474.

[79] K. M. Alam and A. El Saddik, "C2ps: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE Access*, vol. 5, pp. 2050–2062, 2017. DOI: 10.1109/ACCESS.2017.2657006.

[80] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the iot context: A survey on technical features, scenarios, and architectural models," *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1785–1824, Oct. 2020. DOI: 10.1109/JPROC.2020.2998530.

[81] M. A. Taher, H. S. Radhi, and A. K. Jameil, "Enhanced f-ofdm candidate for 5g applications," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 635–652, Jan. 2021. DOI: 10.1007/s12652-020-02046-3.

[82] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," *Transdisciplinary perspectives on complex systems: New findings and approaches*, pp. 85–113, 2017.

[83] Z. Lv and D. Chen, "Improving human living environment and human health through environmental digital twins technology," in *Digital Twin for Healthcare*. Elsevier, 2023, pp. 157–179. DOI: 10.1016/B978-0-32-399163-6.00013-5.

[84] P. Kumar and K. Silambarasan, "Enhancing the performance of healthcare service in iot and cloud using optimized techniques," *IETE Journal of Research*, vol. 68, no. 2, pp. 1475–1484, Mar. 2022. DOI: 10.1080/03772063.2019.1654934.

[85] A. Zaballos *et al.*, "A smart campus' digital twin for sustainable comfort monitoring," *Sustainability*, vol. 12, no. 21, p. 9196, Nov. 2020. DOI: 10.3390/su12219196.

[86] Y. Dai, J. Wang, and S. Gao, "Advanced electronics and artificial intelligence: Must-have technologies toward human body digital twins," *Advanced Intelligent Systems*, vol. 4, no. 7, Mar. 2022. DOI: 10.1002/aisy.202100263.

[87] A. K. Jameil and H. I. Hussein, "Employment compensation capacitor to improve two stage cmos operational amplifier design," *International Journal of Engineering Research & Technology (IJERT)*, vol. 4, no. 4, Apr. 2015. DOI: 10.17577/IJERTV4IS040882.

[88] H. Toloue *et al.*, "Investigation on optical and electrical properties of bilayer graphene," *Diyala Journal of Engineering Sciences*, vol. 8, pp. 535–583, Dec. 2015.

[89] W. Sun *et al.*, "Reducing offloading latency for digital twin edge networks in 6g," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12 240–12 251, Oct. 2020. DOI: 10.1109/TVT.2020.3018817.

[90] Z. Zhang *et al.*, "Artificial intelligence-enabled sensing technologies in the 5g/internet of things era: From virtual reality/augmented reality to the digital twin," *Advanced Intelligent Systems*, vol. 4, no. 7, Mar. 2022. DOI: 10.1002/aisy.202100228.

[91] A. De Benedictis *et al.*, "Digital twins in healthcare: An architectural proposal and its application in a social distancing case study," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 10, pp. 5143–5154, Oct. 2023. DOI: 10.1109/JBHI.2022.3205506.

[92] H. Hassani, X. Huang, and S. MacFeely, "Impactful digital twin in the healthcare revolution," *Big Data and Cognitive Computing*, vol. 6, no. 3, p. 83, Aug. 2022. DOI: 10.3390/bdcc6030083.

[93] P. Thamotharan *et al.*, "Human digital twin for personalized elderly type 2 diabetes management," *Journal of Clinical Medicine*, vol. 12, no. 6, p. 2094, Mar. 2023. DOI: 10.3390/jcm12062094.

[94] M. Maimour, A. Ahmed, and E. Rondeau, "Survey on digital twins for natural environments: A communication network perspective," *Internet of Things*, vol. 25, p. 101 070, Apr. 2024. DOI: 10.1016/j.iot.2024.101070.

[95] A. Abilkaiyrkyzy *et al.*, "Dialogue system for early mental illness detection: Toward a digital twin solution," *IEEE Access*, vol. 12, pp. 2007–2024, 2024. DOI: 10.1109/ACCESS.2023.3348783.

[96] R. A. Mejeed, A. K. Jameil, and H. I. Hussein, "Harmonic amplification damping using a dstatcom-based artificial intelligence controller," *International Journal of Sensors, Wireless Communications and Control*, vol. 9, no. 4, pp. 521–530, Sep. 2019. DOI: 10.2174/2210327909666190611142348.

[97] N. Mohamed *et al.*, "Leveraging digital twins for healthcare systems engineering," *IEEE Access*, vol. 11, pp. 69 841–69 853, 2023. DOI: 10.1109/ACCESS.2023.3292119.

[98] M. Viceconti *et al.*, "Position paper from the digital twins in healthcare to the virtual human twin: A moon-shot project for digital health research," *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 1, pp. 491–501, Jan. 2024. DOI: 10.1109/JBHI.2023.3323688.

[99] Z. Lv, J. Guo, and H. Lv, "Deep learning-empowered clinical big data analytics in healthcare digital twins," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–11, 2024. DOI: 10.1109/TCBB.2023.3252668.

[100] Y. Lin *et al.*, "A novel architecture combining oracle with decentralized learning for iiot," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 3774–3785, Mar. 2023. DOI: 10.1109/JIOT.2022.3150789.

[101] H. Elayan, M. Aloqaily, and M. Guizani, "Digital twin for intelligent context-aware iot healthcare systems," *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 16 749–16 757, Dec. 2021. DOI: 10.1109/JIOT.2021.3051158.

[102] Y. Zhong *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)*, Piscataway, NJ, USA: IEEE, Dec. 2022, pp. 21–26. DOI: 10.1109/ICIR55739.2022.00020.

[103] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," White paper, Tech. Rep., 2014.

[104] F. Tao *et al.*, "Digital twin driven prognostics and health management for complex equipment," *CIRP Annals*, vol. 67, no. 1, pp. 169–172, 2018. DOI: 10.1016/j.cirp.2018.04.055.

[105] G. Coorey, G. A. Figtree, *et al.*, "The health digital twin to tackle cardiovascular disease—a review of an emerging interdisciplinary field," *NPJ Digital Medicine*, vol. 5, no. 1, p. 126, Aug. 2022. DOI: 10.1038/s41746-022-00640-7.

[106] Q. Yang *et al.*, "Development of digital fetal heart models with virtual ultrasound function based on cardiovascular casting and computed tomography scan," *Bioengineering*, vol. 9, no. 10, p. 524, Oct. 2022. DOI: 10.3390/bioengineering9100524.

[107] K. Gillette *et al.*, "A framework for the generation of digital twins of cardiac electrophysiology from clinical 12-leads ecgs," *Medical Image Analysis*, vol. 71, p. 102 080, Jul. 2021. DOI: 10.1016/j.media.2021.102080.

[108] W. Wang *et al.*, "Digital twin rehabilitation system based on self-balancing lower limb exoskeleton," *Technology and Health Care*, vol. 31, no. 1, pp. 103–115, Jan. 2023. DOI: 10.3233/THC-220087.

[109] Y. Liu *et al.*, "A novel cloud-based framework for the elderly healthcare services using digital twin," *IEEE Access*, vol. 7, pp. 49 088–49 101, 2019. DOI: 10.1109/ACCESS.2019.2909828.

[110] H. Kasani *et al.*, "Fabrication of carbon nanoparticle / polymer nanocomposite based thermometer," *Diyala Journal of Engineering Sciences*, vol. 8, pp. 606–621, Dec. 2015.

[111] "Max30102, high-sensitivity pulse oximeter and heart-rate sensor for wearable health." [Online; accessed 16-Jul-2024]. (2024), [Online]. Available: https://www.analog.com/en/products/max30102.html.

[112] *Digital plug & play infrared thermometer in a to-can*, Melexis, 2024.

[113] R. Ferdousi *et al.*, "Digital twins for well-being: An overview," *Digital Twin*, vol. 1, p. 7, Feb. 2022. DOI: 10.12688/digitaltwin.17475.2.

[114] F. Laamarti *et al.*, "An iso/ieee 11073 standardized digital twin framework for health and well-being in smart cities," *IEEE Access*, vol. 8, pp. 105 950–105 961, 2020. DOI: 10.1109/ACCESS.2020.2999871.

[115] Y.-H. Chen *et al.*, "14.5 eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA: IEEE, Jan. 2016, pp. 262–263. DOI: 10.1109/ISSCC.2016.7418007.

[116] H. A. Rivera *et al.*, "Towards continuous monitoring in personalized healthcare through digital twins," in *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, 2019, pp. 329–335.

[117] T. Erol, A. F. Mendi, and D. Dogan, "The digital twin revolution in healthcare," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Istanbul, Turkey: IEEE, Oct. 2020, pp. 1–7. DOI: 10.1109/ISMSIT50672.2020.9255249.

[118] R. Martinez-Velazquez, R. Gamez, and A. El Saddik, "Cardio twin: A digital twin of the human heart running on the edge," in *2019 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, Istanbul, Turkey: IEEE, Jun. 2019, pp. 1–6. DOI: 10.1109/MeMeA.2019.8802162.

[119] S. Agostinelli *et al.*, "The potential of digital twin model integrated with artificial intelligence systems," in *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I& CPS Europe)*, Madrid, Spain: IEEE, Jun. 2020, pp. 1–6. DOI: 10.1109/EEEIC/ICPSEurope49358.2020.9160810.

[120] Y. Lu, W. Liu, and H. Cui, "Msa vs. mvc: Future trends for big data processing platforms," in *Smart Computing and Communication*, M. Qiu, Ed. Cham: Springer International Publishing, 2018, vol. 10699, pp. 310–320. DOI: 10.1007/978-3-319-73830-7_31.

[121] B. R. Barricelli *et al.*, "Human digital twin for fitness management," *IEEE Access*, vol. 8, pp. 26 637–26 664, 2020. DOI: 10.1109/ACCESS.2020.2971576.

[122] H.-J. Jiang, Y.-A. Huang, and Z.-H. You, "Saerof: An ensemble approach for large-scale drug-disease association prediction by incorporating rotation forest and sparse autoencoder deep neural network," *Scientific Reports*, vol. 10, no. 1, p. 4972, Mar. 2020. DOI: 10.1038/s41598-020-61616-9.

[123] J. Zhang *et al.*, "Cyber resilience in healthcare digital twin on lung cancer," *IEEE Access*, vol. 8, pp. 201 900–201 913, 2020. DOI: 10.1109/ACCESS.2020.3034324.

[124] J. Čuklina, P. G. A. Pedrioli, and R. Aebersold, "Review of batch effects prevention, diagnostics, and correction approaches," in *Mass Spectrometry Data Analysis in Proteomics*, R. Matthiesen, Ed. New York, NY: Springer New York, 2020, vol. 2051, pp. 373–387. DOI: 10.1007/978-1-4939-9744-2_16.

[125] S. Shahrivari, "Beyond batch processing: Towards real-time and streaming big data," *Computers*, vol. 3, no. 4, pp. 117–129, Oct. 2014. DOI: 10.3390/computers3040117.

[126] N. Deepa *et al.*, "An ai-based intelligent system for healthcare analysis using ridge-adaline stochastic gradient descent classifier," *The Journal of Supercomputing*, vol. 77, no. 2, pp. 1998–2017, Feb. 2021. DOI: 10.1007/s11227-020-03347-2.

[127] A. Karakra *et al.*, "Hospit'win: A predictive simulation-based digital twin for patients pathways in hospital," in *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, Chicago, IL, USA: IEEE, May 2019, pp. 1–4. DOI: 10.1109/BHI.2019.8834534.

[128] "Digital twin approach to clinical dss with explainable ai: Reducing subjectivity in liver disease diagnosis using domain knowledge and machine learning." [Online; accessed 16-Jul-2024]. (2019), [Online]. Available: https://colab.research.google.com/drive/1Q3Eop0e26_S2QueGH.

[129] C. Ossai and N. Wickramasinghe, "A bayesian network model to establish a digital twin architecture for superior falls risk prediction," *AMCIS 2021 Proceedings*, Aug. 2021.

[130] R. G. Diaz, F. Laamarti, and A. El Saddik, "Dtcoach: Your digital twin coach on the edge during covid-19 and beyond," *IEEE Instrumentation & Measurement Magazine*, vol. 24, no. 6, pp. 22–28, Sep. 2021. DOI: 10.1109/MIM.2021.9513635.

[131] K. Bruynseels, F. Santoni De Sio, and J. Van Den Hoven, "Digital twins in health care: Ethical implications of an emerging engineering paradigm," *Frontiers in Genetics*, vol. 9, p. 31, Feb. 2018. DOI: 10.3389/fgene.2018.00031.

[132] "Explainable artificial intelligence (xai)." [Online; accessed 16-Jul-2024], Defense Advanced Research Projects Agency (DARPA). (2024), [Online]. Available: https://www.darpa.mil/program/explainable-artificial-intelligence.

[133] M. Nazari *et al.*, "Explainable ai to improve acceptance of convolutional neural networks for automatic classification of dopamine transporter spect in the diagnosis of clinically uncertain parkinsonian syndromes," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 49, no. 4, pp. 1176–1186, Mar. 2022. DOI: 10.1007/s00259-021-05569-9.

[134] N. K. Chakshu, I. Sazonov, and P. Nithiarasu, "Towards enabling a cardiovascular digital twin for human systemic circulation using inverse analysis," *Biomechanics and Modeling in Mechanobiology*, vol. 20, no. 2, pp. 449–465, Apr. 2021. DOI: 10.1007/s10237-020-01393-6.

[135] D. Drummond, J. Roukema, and M. Pijnenburg, "Home monitoring in asthma: Towards digital twins," *Current Opinion in Pulmonary Medicine*, vol. 29, no. 4, pp. 270–276, Jul. 2023. DOI: 10.1097/MCP.0000000000000963.

[136] G. Cappon *et al.*, "Replaybg: A digital twin-based methodology to identify a personalized model from type 1 diabetes data and simulate glucose concentrations to assess alternative therapies," *IEEE Transactions on Biomedical Engineering*, vol. 70, no. 11, pp. 3227–3238, Nov. 2023. DOI: 10.1109/TBME.2023.3286856.

[137] R. Yamaganti, P. N. S. Jyothi, and S. U. Manjari, "The role of internet of things in developing competitive healthcare devices: A case study in the digital healthcare industry," in *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, Coimbatore, India: IEEE, Feb. 2023, pp. 82–86. DOI: 10.1109/ICAIS56108.2023.10073802.

[138] Y. A. Abbas *et al.*, "A systematic mapping study on radio frequency identification security," *International Journal of Sensors, Wireless Communications and Control*, vol. 10, no. 5, pp. 659–668, Feb. 2021. DOI: 10.2174/2210327909666191004095834.

[139] T. Patel and A. K. Pandey, "Advanced medical nursing care unit for emergency healthcare based on the internet of things," in *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, Greater Noida, India: IEEE, Jan. 2023, pp. 1428–1432. DOI: 10.1109/AISC56616.2023.10085394.

[140] L. Mhamdi and H. Abdul Khalek, "Congestion control in constrained internet of things networks," *IET Wireless Sensor Systems*, vol. 13, no. 6, pp. 247–255, Dec. 2023. DOI: 10.1049/wss2.12072.

[141] W. J. Benny *et al.*, "Comprehensive iot-based monitoring system for improving worker's health and safety at the high-risk working environment," in *2024 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, Miri Sarawak, Malaysia: IEEE, Jan. 2024, pp. 87–93. DOI: 10.1109/GECOST60902.2024.10474774.

[142] A. Alqahtani, S. Alsubai, and M. Bhatia, "Digital-twin-assisted healthcare framework for adult," *IEEE Internet of Things Journal*, vol. 11, no. 8, pp. 14 963–14 970, Apr. 2024. DOI: 10.1109/JIOT.2023.3345331.

[143] A. Vallée, "Envisioning the future of personalized medicine: Role and realities of digital twins," *Journal of Medical Internet Research*, vol. 26, e50204, May 2024. DOI: 10.2196/50204.

[144] Y. Almadany *et al.*, "A novel algorithm for estimation of twitter users location using public available information," *International Journal on Smart Sensing and Intelligent Systems*, vol. 13, no. 1, pp. 1–10, Jan. 2020. DOI: 10.21307/ijssis-2020-012.

[145] M. C. Hlophe and B. T. Maharaj, "From cyber–physical convergence to digital twins: A review on edge computing use case designs," *Applied Sciences*, vol. 13, no. 24, p. 13 262, Dec. 2023. DOI: 10.3390/app132413262.

[146] E. Faliagka *et al.*, "Trends in digital twin framework architectures for smart cities: A case study in smart mobility," *Sensors*, vol. 24, no. 5, p. 1665, Mar. 2024. DOI: 10.3390/s24051665.

[147] X. Li *et al.*, "Big data analysis of the internet of things in the digital twins of smart city based on deep learning," *Future Generation Computer Systems*, vol. 128, pp. 167–177, Mar. 2022. DOI: 10.1016/j.future.2021.10.006.

[148] D. Menon, B. Anand, and C. L. Chowdhary, "Digital twin: Exploring the intersection of virtual and physical worlds," *IEEE Access*, vol. 11, pp. 75 152–75 172, 2023. DOI: 10.1109/ACCESS.2023.3294985.

[149] P. Jia, X. Wang, and X. Shen, "Accurate and efficient digital twin construction using concurrent end-to-end synchronization and multi-attribute data resampling," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 4857–4870, Mar. 2023. DOI: 10.1109/JIOT.2022.3221012.

[150] O. Kocabas, T. Soyata, and M. K. Aktas, "Emerging security mechanisms for medical cyber physical systems," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 3, pp. 401–416, May 2016. DOI: 10.1109/TCBB.2016.2520933.

[151] T. Zhu *et al.*, "Glugan: Generating personalized glucose time series using generative adversarial networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 10, pp. 5122–5133, Oct. 2023. DOI: 10.1109/JBHI.2023.3271615.

[152] S. Sarp *et al.*, "Digital twin in healthcare: A study for chronic wound management," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 11, pp. 5634–5643, Nov. 2023. DOI: 10.1109/JBHI.2023.3299028.

[153] Y. Yang *et al.*, "Dynamic human digital twin deployment at the edge for task execution: A two-timescale accuracy-aware online optimization," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2024. DOI: 10.1109/TMC.2024.3406607.

[154] M. S. Dihan *et al.*, "Digital twin: Data exploration, architecture, implementation and future," *Heliyon*, vol. 10, no. 5, e26503, Mar. 2024. DOI: 10.1016/j.heliyon.2024.e26503.

[155] M. Sheraz *et al.*, "A comprehensive survey on revolutionizing connectivity through artificial intelligence-enabled digital twin network in 6g," *IEEE Access*, pp. 1–1, 2024. DOI: 10.1109/ACCESS.2024.3384272.

[156] S. Qiu *et al.*, “Improved binary marine predator algorithm-based digital twin-assisted edge-computing offloading method,” *Future Generation Computer Systems*, vol. 155, pp. 437–446, Jun. 2024. DOI: 10.1016/j.future.2024.02.021.

[157] E. Bozkaya *et al.*, “Proof of evaluation-based energy and delay aware computation offloading for digital twin edge network,” *Ad Hoc Networks*, vol. 149, p. 103 254, Oct. 2023. DOI: 10.1016/j.adhoc.2023.103254.

[158] L. Chen *et al.*, “A3c-based and dependency-aware computation offloading and service caching in digital twin edge networks,” *IEEE Access*, vol. 11, pp. 57 564–57 573, 2023. DOI: 10.1109/ACCESS.2023.3284461.

[159] T. Liu *et al.*, “Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network,” *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1427–1444, Jan. 2022. DOI: 10.1109/JIOT.2021.3086961.

[160] H. B. Eldeeb *et al.*, “Digital twin-assisted owc: Towards smart and autonomous 6g networks,” *IEEE Network*, pp. 1–1, 2024. DOI: 10.1109/MNET.2024.3374370.

[161] Y. Sun *et al.*, “Prediction model of in-hospital mortality in intensive care unit patients with cardiac arrest: A retrospective analysis of mimic -iv database based on machine learning,” *BMC Anesthesiology*, vol. 23, no. 1, p. 178, May 2023. DOI: 10.1186/s12871-023-02138-5.

[162] B. Erdoğan and H. Oğul, “Objective pain assessment using vital signs,” *Procedia Computer Science*, vol. 170, pp. 947–952, 2020. DOI: 10.1016/j.procs.2020.03.103.

[163] F. Michard, K. Shelley, and E. L’Her, “Covid-19: Pulse oximeters in the spotlight,” *Journal of Clinical Monitoring and Computing*, vol. 35, no. 1, pp. 11–14, Feb. 2021. DOI: 10.1007/s10877-020-00550-7.

[164] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, 1st ed. Cambridge University Press, May 2014. DOI: 10.1017/CBO9781107298019.

[165] D. Zhang and Y. Gong, “The comparison of lightgbm and xgboost coupling factor analysis and prediagnosis of acute liver failure,” *IEEE Access*, vol. 8, pp. 220 990–221 003, 2020. DOI: 10.1109/ACCESS.2020.3042848.

[166] G. Bryson and D. O’Dwyer, “Benefits and challenges of digital pathology use for primary diagnosis in gynaecological practice: A real-life experience,” *Diagnostic Histopathology*, Jul. 2023. DOI: 10.1016/j.mpdhp.2023.07.001.

[167] S. S. Akash and M. S. Ferdous, “A blockchain based system for healthcare digital twin,” *IEEE Access*, vol. 10, pp. 50 523–50 547, 2022. DOI: 10.1109/access.2022.3173617.

[168] K. Nsafoa-Yeboah *et al.*, "Flexible open network operating system architecture for implementing higher scalability using disaggregated software-defined optical networking," *IET Networks*, Dec. 2023. DOI: 10.1049/ntw2.12110.

[169] Z. Wenhua *et al.*, "Data security in smart devices: Advancement, constraints and future recommendations," *IET Networks*, vol. 12, no. 6, pp. 269–281, Jun. 2023. DOI: 10.1049/ntw2.12091.

[170] B. Hammi *et al.*, "Iot technologiesfor smart cities," *IET Networks*, vol. 7, no. 1, pp. 1–13, Jan. 2018. DOI: 10.1049/iet-net.2017.0163.

[171] V. F. Rodrigues *et al.*, "Digital health in smart cities: Rethinking the remote health monitoring architecture on combining edge, fog, and cloud," *Health and Technology*, pp. 1–24, 2023.

[172] O. B. J. Rabie *et al.*, "A full privacy-preserving distributed batch-based certificate-less aggregate signature authentication scheme for healthcare wearable wireless medical sensor networks (HWMSNs)," *International Journal of Information Security*, vol. 23, no. 1, pp. 51–80, Feb. 2024. DOI: 10.1007/s10207-023-00748-1.

[173] B. Wang *et al.*, "Human digital twin in the context of industry 5.0," *Robotics and Computer-Integrated Manufacturing*, vol. 85, p. 102 626, Feb. 2024. DOI: 10.1016/j.rcim.2023.102626.

[174] C. Das *et al.*, "Toward iort collaborative digital twin technology enabled future surgical sector: Technical innovations, opportunities and challenges," *IEEE Access*, vol. 10, pp. 129 079–129 104, 2022. DOI: 10.1109/access.2022.3227644.

[175] R. M. Soares *et al.*, "Digital twin for monitoring of industrial multi-effect evaporation," *Processes*, vol. 7, no. 8, p. 537, Aug. 2019. DOI: 10.3390/pr7080537.

[176] Y. Feng, "Create the individualized digital twin for noninvasive precise pulmonary healthcare," *Significances of Bioengineering & Biosciences*, vol. 1, no. 2, Jan. 2018. DOI: 10.31031/sbb.2018.01.000507.

[177] O. C. Madubuike and C. J. Anumba, "Digital twin–based health care facilities management," *Journal of Computing in Civil Engineering*, vol. 37, no. 2, p. 04 022 057, Mar. 2023. DOI: 10.1061/JCCEE5.CPENG-4842.

[178] J.-L. Aufranc, *Espressif rolls out esp32 boards for microsoft azure iot*, https://www.cnx-software.com/2019/05/09, Accessed: May, 9, 2019, 2019.