

Edge-based intelligent and smart health monitoring on PYNQ-Z2 using lightweight protocol and integration of machine learning models

Samik Basu¹, Rajdeep Ray^{2*},
Arkadip Maitra³, Pritha Banerjee¹
and Amlan Chakrabarti⁴

¹Department of Computer Science
and Engineering, University of
Calcutta, Kolkata 700106, India

²Department of Electronics and
Communication Engineering,
Swami Vivekananda University,
Kolkata 700121, India

³Department of Computer Science,
Ramakrishna Mission Vivekananda
Educational and Research Institute
(RKMVERI) Belur Math, Howrah
711202, India

⁴A. K. Choudhury School of
Information Technology, University
of Calcutta, Kolkata 700106, India

*E-mail: ray.rajdeep78@gmail.com

Received for publication
June 29, 2025.

Abstract

This study presents a smart health monitoring system that integrates information and communication technology (ICT) with machine learning (ML) for efficient, real-time healthcare delivery. The system is designed for elderly and chronically ill patients, using sensor-based data acquisition (heart rate [HR], blood oxygen saturation [SpO₂], temperature) and self-powered, remote monitoring via personal health devices (PHDs). It employs the message queuing telemetry transport (MQTT) protocol for lightweight messaging, with sensor data collected through an ESP32-based microcontroller, working as an MQTT client, and a Raspberry Pi (RPI) acting as the MQTT broker, enabling seamless health data accessibility over the internet. An intelligent health prediction module is designed to classify health conditions based on key physiological parameters, such as HR, SpO₂, and body temperature. Health conditions are classified using ML algorithms (support vector machine [SVM], K-nearest neighbors [KNN], decision tree [DT]) trained on real-time datasets. The models are deployed on a PYNQ-Z2 FPGA board, ensuring high-speed, low-power, and cost-effective operation. This scalable and intelligent framework supports accessible and reliable e-health monitoring.

Keywords

ESP32, MQTT broker, Raspberry Pi-3, sensors, IoT, machine learning algorithms, PYNQ-Z2

Abbreviations

ADCs, Analog-to-Digital Converters; API, Application Programming Interface; ARM, Advanced RISC Machine; AWS, Amazon Web Services; AXI, Advanced eXtensible Interface; BR, Breathing Rate; BRAM, Block Random Access Memory; CNN, Convolutional Neural Network; DDR, Double Data Rate; DMA, Direct Memory Access; DSP, Digital Signal Processor/Processing; ECG, Electrocardiogram; EEG, Electroencephalogram; FPGA, Field-Programmable Gate Array; FPGA-SoCs, Field-Programmable Gate Array System-on-Chips; GSM, Global System for Mobile Communications; IDE, Integrated Development Environment; JTAG, Joint Test Action Group; LCD, Liquid Crystal Display; LEDs, Light Emitting Diodes; SCL, Serial Clock Line; SDA, Serial Data Line; SoC, System-on-Chip; SPI, Serial Peripheral Interface; UART, Universal Asynchronous Receiver-Transmitter; ZYNQ, Xilinx Zynq (ARM-FPGA based SoC platform); ZYNQ-SoC, Xilinx Zynq System-on-Chip.

I. Introduction

In the current era, the growing global population and inadequate healthcare infrastructure have led to an increased reliance on self-healthcare systems. Escalating healthcare costs and unhealthy lifestyles have contributed to a surge in chronic diseases such as diabetes, hypertension, stroke, and cardiovascular disorders. Continuous patient monitoring and early detection of chronic conditions are vital to mitigating their fatal consequences. The Internet of Things (IoT) has emerged as a transformative technology for remote patient monitoring [1, 2]. Personal health devices (PHDs) leveraging IoT capabilities can collect critical patient data, such as temperature, blood oxygen saturation (SpO_2) level, and heart rate (HR), and facilitate real-time communication with healthcare providers [18–20]. By enabling healthcare services over long distances, IoT reduces hospital stays, travel costs, and associated healthcare expenses [14]. The message queuing telemetry transport (MQTT) protocol plays a pivotal role in enabling efficient Machine-to-Machine (M2M) communication [15]. MQTT, a lightweight messaging protocol based on a publish-subscribe pattern, ensures efficient message transmission between devices [34]. Its key feature is the ability to buffer unsent messages during a broker disconnection and deliver them upon reconnection, ensuring reliable communication in constrained network environments.

The objective of this study is to develop a portable self-health monitoring system that can operate conveniently at home or in rural and remote areas. The system facilitates remote communication using an IoT platform, providing access to medical expertise irrespective of geographical constraints. By integrating machine learning (ML) algorithms, the system offers advanced predictive capabilities, enabling early detection and management of chronic diseases. The proposed intelligent health monitoring system measures critical physiological parameters such as HR, SpO_2 , and body temperature. Additionally, it integrates the MQTT protocol and IoT platform to enable disease diagnosis and facilitate timely interventions by healthcare professionals. In emergency scenarios, the system alerts doctors and caregivers. Even in cases of unavailability of remote doctor access, the system employs ML algorithms to intelligently predict patient conditions based on recorded data, ensuring uninterrupted monitoring and proactive management.

This paper is structured as follows: Section II provides a literature survey, while Section III details the proposed system architecture. Section IV elaborates data collection and communication. Section

V describes the health prediction model utilizing ML classifiers. The working principle of the proposed system is elaborated in Section VI. Section VII discusses the experimental setup, results, and hardware testing. Finally, Section VIII concludes the paper.

II. Literature Survey

IoT refers to a networked ecosystem where various devices, including portable, wearable, and household devices, are interconnected and integrated with software and hardware, enabling seamless communication across networks. In healthcare, IoT facilitates real-time monitoring, offering flexibility and enabling vital health functions to be observed remotely at any time and place. This capability ensures timely data transmission and patient status monitoring, significantly improving healthcare outcomes [1–4].

IoT-based pervasive monitoring systems consist of two primary components: data collection and data transmission. These systems typically follow a three-layer architecture: the sensing layer, the transport layer, and the application layer [8]. Numerous researches have focused on real-time health monitoring, enabling remote medical services and patient care based on IoT and ML [29–31]. Tonello et al. [5] introduced a sensing platform tailored for remote healthcare applications targeting elderly people. Sharma et al. [6] proposed a novel concept called the Schizophrenia Detection Cap (SczCap), an easy-to-wear cap that integrates both hardware and software to acquire EEG signals from the scalp for precise schizophrenia detection. Zhang et al. [7] presented a novel portable wearable cardiorespiratory sensor integrated with a smartphone, offering a compact and efficient solution for comprehensive cardiorespiratory monitoring, which enables precise measurement of respiratory rate (RR), HR, and heart rate variability (HRV). Wang et al. [28] introduced an outdoor monitoring system for elderly individuals, capable of transmitting physiological signals and fall event data to a healthcare center anytime, from any location. Gupta et al. [32] proposed a system using the Intel Galileo 2nd board for temperature and HR monitoring, though it lacked ECG integration, a critical feature for elderly care.

Pardeshi et al. [9] designed a health monitoring system that captured abnormalities in parameters, such as blood pressure, temperature, and ECG, using GSM or Wi-Fi for alerts. Divakaran et al. [10] created a multi-sensor system with live video feeds and parameter reporting to aid doctors in monitoring patient status remotely. Uddin et al. [11] focused on intensive care unit (ICU) monitoring, enabling specialists to track multiple patients in real time through

cloud-connected sensors. Furthermore, various ML-based systems have emerged, integrating IoT frameworks for advanced health monitoring [33, 35, 47]. Bharadwaj et al. [29] introduced both a comprehensive compilation and a review of the state-of-the-art applications of ML algorithms currently being integrated with Healthcare Internet-of-Things (H-IoT) systems. Hussain et al. [24] presented accelerated K-nearest neighbors (KNN) classifiers on FPGA for ensemble classification, while Vieira et al. [25] proposed a flexible streaming KNN classifier for embedded FPGA-SoCs. Other studies, such as those by Samiee et al. [26] and Danopoulos et al. [27], aimed to enhance KNN classification through optimization techniques. Recent IoT systems, such as those by Basu et al. [18–20] and Jana et al. [21, 22], have focused on non-invasive health parameter monitoring and coronavirus disease-2019 (COVID-19) tracking, respectively. Chooruang and Mangkalakeeree [23] introduced an integrated MQTT protocol for lightweight communication in HR monitoring systems. By contrast, Bouslama et al. [12] adopted AWS for real-time alerts without data storage, and Stradolini et al. [13] introduced smartwatch-based monitoring for critical patient management. The summary of the literature survey in tabular format is shown in Table 1.

III. Proposed System Architecture

This study aims to develop an IoT-based healthcare system capable of collecting multiple physiological parameters from a patient, analyzing the data to detect abnormalities, and transmitting the information to a server for storage. The system also automatically notifies the physician via email about the patient's health status and displays the measured parameters on a web interface. Additionally, the system employs ML algorithms to predict the patient's health condition based on the collected data. This section outlines the essential components of the proposed system, including data collection and communication mechanisms, as well as the ML models utilized for health condition analysis.

a. Necessary components

a.i HR and pulse oximeter sensor (MAX30100)

The MAX30100 is an integrated sensor that combines pulse oximetry and HR monitoring capabilities [16]. It features two LEDs, a photodetector, and low-noise analog signal processing. The sensor operates on the

principle of remote photo plethysmography (rPPG), which measures HR and SpO₂ levels by analyzing the varying absorption of red and infrared lights through oxygenated and deoxygenated blood. For accurate results, measurements are typically taken from the fingertip, where capillaries are near the surface. The sensor's optical reflection method is ideal for wearable applications, making it a reliable choice for this system.

a.ii Body temperature sensor (DS18B20)

The DS18B20 digital temperature sensor provides temperature readings with a resolution ranging from 9 bits to 12 bits in Celsius [37]. Its design allows for measurements from various locations such as the underarm, oral cavity, or ear canal. The sensor communicates using a single data line, making it efficient and easy to interface with the microcontroller.

a.iii ESP32 microcontroller

The ESP32 is a powerful dual-core 32-bit microcontroller with built-in Wi-Fi and Bluetooth modules [36]. Its low-cost, energy-efficient design includes features such as 4 MB flash memory, 520 KB SRAM, multiple I2C, SPI, and UART interfaces, and 12-bit ADCs. This makes it ideal for sensor integration, data processing, and communication tasks.

a.iv Raspberry Pi (RPI)

The RPI is a compact microcomputer with Linux-based Raspbian OS [17]. It serves as a cost-effective platform for interfacing with IoT devices. The RPI 3 Model B used in this system includes a quad-core processor, 1 GB RAM, built-in Wi-Fi, Bluetooth low energy (BLE), and a variety of input/output ports. It acts as an MQTT broker, facilitating data transfer and storage.

IV. Data Collection and Communication

The proposed system comprises three primary subsystems: data collection, data acquisition, and communication. Each subsystem is outlined below.

a. Data collection

This subsystem gathers multiple medical parameters such as HR, body temperature, SpO₂, and vital signs using sensors. Analog data from the sensors are processed and converted to a digital format for storage

Table 1: Literature survey summary

Study/authors	Key focus/contribution	Key features/applications	Year
IoT in Healthcare [1–4]	Real-time health monitoring with IoT, enabling remote patient	Flexible monitoring at any time and place	2015, 2021, 2023, 2018
Tonello et al. [5]	Sensing platform for remote healthcare	Targeted at elderly care	2024
Sharma et al. [6]	Schizophrenia Detection Cap (SczCap)	Wearable cap for EEG signal acquisition and precise schizophrenia detection	2023
Zhang et al. [7]	Portable wearable cardiorespiratory sensor	Integrated with a smartphone; measures RR and HR	2024
Priya et al. [8]	IoT-based pervasive monitoring systems	Three-layer architecture: sensing layer, transport layer, and application layer	2019
Pardeshi et al. [9]	Health monitoring system for detecting abnormalities	Captures blood pressure, temperature, and ECG data; uses GSM or Wi-Fi for alerts	2017
Divakaran et al. [10]	Multi-sensor system with live video feeds and parameter reporting for remote patient monitoring	Enhanced doctor–patient communication	2017
Uddin et al. [11]	ICU monitoring with cloud-connected sensors for real-time tracking	Supported simultaneous monitoring of multiple patients	2017
Bousslama et al. [12]	Adopted AWS for real-time alerts without data	Reduced dependency on local storage resources	2019
Stradolini et al. [13]	Smartwatch-based monitoring	Critical patient management	2017
Basu et al. [18–20]	Developed IoT systems for non-invasive health parameter monitoring	Addressed healthcare needs in wearable systems. Focused on recent IoT systems	2020
Jana et al. [21–22]	Focused on IoT-based COVID-19 tracking	Non-invasive and effective tracking during the pandemic	2021, 2022
Chooruang and Mangkalakeeree [23]	Integrated MQTT protocol for lightweight communication in HR monitoring systems	Simplified communication and reduced latency	2016
Hussain et al. [24]	Accelerated KNN classifiers on FPGA for ensemble classification	Enhanced classification performance in IoT frameworks	2012
Vieira et al. [25]	Proposed flexible streaming KNN classifier for embedded FPGA-SoCs	Optimized for IoT systems	2019
Samiee et al. [26] and Danopoulos et al. [27]	Focused on enhancing KNN classification through optimization	Improved classification accuracy and speed	2018, 2019
Wang et al. [28]	Outdoor monitoring system for elderly individuals	Transmits physiological signals and fall events to healthcare centers	2016
Bharadwaj et al [29]	Review of ML algorithms in H-IoT systems	Compilation of state-of-the-art ML applications in IoT healthcare	2021
Gupta et al. [32]	System using Intel Galileo 2nd board for temperature and HR monitoring	Monitors temperature and HR; lacks ECG integration	2016

COVID-19, coronavirus disease-2019; H-IoT, healthcare Internet of things; HR, heart rate; ICU, intensive care unit; IoT, Internet of Things; KNN, K-nearest neighbors; ML, machine learning; MQTT, message queuing telemetry transport; RR, respiratory rate; SoC, system-on-chip.

and transmission. Table 2 presents and summarizes the dataset for the proposed system.

- (1) **HR and blood oxygen monitoring:** The MAX30100 sensor measures HR and SpO_2 levels through optical reflection. It uses the I2C communication protocol via SDA and SCL pins to interface with the ESP32 microcontroller.
- (2) **Body temperature measurement:** The DS18B20 sensor measures body temperature using methods such as oral, axillary, or tympanic. It communicates with the ESP32 via the SPI protocol for configuration and data transfer.

b. Data acquisition

The ESP32 microcontroller is used for data acquisition, processing, and transmission. It supports low-power operations and is programmable using Arduino IDE or Micro-Python. Equipped with multiple communication interfaces (I2C, SPI, and UART), it processes sensor data and performs initial analysis to detect abnormalities.

c. Communication

The communication subsystem is designed for fast, low-cost, and reliable data transfer. The ESP32 handles data transmission using the MQTT protocol, while the RPI serves as the MQTT broker for data storage and management.

- (1) **ESP32 with MQTT:** The ESP32 connects to a Wi-Fi network and acts as an MQTT client to publish sensor data to specific topics (e.g., esp32/pulse, esp32/temperature, and esp32/ SpO_2). The MQTT broker, running on RPI, receives and manages these messages.
- (2) **RPI as MQTT broker:** The RPI, running Raspbian OS, hosts the MQTT broker (Mosquitto) to facilitate data exchange between devices. It stores patient data, provides historical information, and acts as a web server for remote monitoring. The MQTT protocol ensures reliable message delivery and supports thousands of clients, making it ideal for scalable healthcare applications.

The system's MQTT architecture employs a publish/subscribe model for efficient data exchange. Users can access the monitoring system remotely by entering the server's IP address. Based on login credentials, the system provides specific functionalities for administrators, doctors, and patients. For instance, doctors can monitor patient health, search records, and manage patient lists via a dedicated web interface.

V. Design of Health Prediction Unit Using ML Classifiers

The intelligent health prediction unit leverages ML techniques to assess a patient's heart health, SpO_2

Table 2: Dataset for the proposed system

Index	Subject	Session	Age group	Day	HR (bpm)	SpO_2	Temp ($^{\circ}\text{F}$)
0	S1	Morning (rest)	A	Day 1	79	98	103
1	S1	Morning (rest)	A	Day 2	76	97	98
2	S1	Morning (rest)	A	Day 3	74	97	90
3	S1	Morning (rest)	A	Day 4	75	98	100
4	S1	Morning (rest)	A	Day 5	72	96	92
...	...	...	...	...	...	...	...
895	S10	Evening (normal)	C	Day 6	81	98	110
896	S10	Evening (normal)	C	Day 7	79	96	108
897	S10	Evening (normal)	C	Day 8	86	96	90
898	S10	Evening (normal)	C	Day 9	91	81	95
899	S10	Evening (normal)	C	Day 10	87	96	97

HR, heart rate; SpO_2 , blood oxygen saturation.

levels, and body temperature based on health parameters, gender, and age. The system is trained using real-time data collected from the DS18B20 and MAX30100 sensors. For this implementation, a support vector machine (SVM), KNN, and decision tree (DT) algorithms are employed for training and testing. The training dataset comprises real-time data acquired through the proposed system and incorporates classifications aligned with the American Heart Association (AHA) standards for cardiac health. This prediction unit provides critical insights into a patient's condition, especially in the absence of immediate medical advice. It also facilitates efficient preliminary diagnoses, thereby reducing the workload of health-care professionals and enabling rapid, accessible health assessments.

a. Overview of different classifiers

The system employs various ML classifiers, including KNN, SVM, and DT. A brief overview of these classifiers is presented below [41]:

- (1) **KNN algorithm:** KNN [38] is a nonparametric supervised learning algorithm that classifies new data points based on their similarity to the K-nearest training data points. The algorithm calculates the distance between the new data point and all training data points and assigns the majority class label among the KNN.

Mathematical representation

- (a) Let k be the number of neighbors considered for classification.
 - (b) $D(x_i, x_j)$: A distance function between data points x_i and x_j .
 - (c) For a new instance x , the algorithm identifies its KNN and assigns it the majority class label.
- (2) **SVM algorithm:** SVM [39] constructs a hyperplane to separate data points into distinct classes. The algorithm seeks the hyperplane that maximizes the margin, defined as the distance between the hyperplane and the nearest data points from either class. A linear kernel is used in this system to simplify computation.

Mathematical representation

- (a) Let X represent the training data points and Y the class labels.

- (b) The hyperplane is defined as $w \cdot x + b = 0$, where w is the weight vector and b the bias.
- (c) The SVM maximizes the margin while ensuring data points on the same side of the hyperplane are in the same class.

- (3) **DT algorithm:** The DT [40] algorithm uses a tree-like structure to classify data points. Each internal node represents a feature, branches denote decisions based on feature values, and leaf nodes represent the final class labels. The tree is constructed by recursively splitting the data based on the most significant feature.

Mathematical representation

- (a) Let X be the training data points and Y the class labels.
- (b) At each node, split X into subsets X_l and X_r based on a feature f and threshold θ :
 - $X_l = \{x \in X \mid x[f] < \theta\}$
 - $X_r = \{x \in X \mid x[f] \geq \theta\}$
- (c) Recursively, the trees are built until each leaf node contains data points of the same class.

b. Detailed steps of ML algorithms

The health monitoring system employs the following key steps for ML-based prediction:

- (1) **Data collection and preparation:** Data were collected from the proposed hardware system, which measured various health parameters. This dataset, comprising more than 900 complete patient records, was used to train and validate the model. Data collection timing varied, ensuring a diverse range of samples. Key criteria for data selection included:
 - Inclusion of critical health parameters such as HR, SpO₂, and temperature.
 - Consistent formatting and completeness of data records.
- (2) **Feature extraction and classification:** The collected data were processed to extract relevant features for classification. These features included:
 - Vital signs (HR, SpO₂, and body temperature)
 - Demographic information (gender and age)

The extracted features were fed into the ML classifiers (KNN, SVM, and DT) to train the prediction model. Once trained, the system classified patient health status and identified potential abnormalities.

(3) **Evaluation metrics:** To evaluate the reliability of the ML algorithms trained on the dataset, we used four key metrics: accuracy, precision, recall, and the F1 score. These metrics [42] provide a comprehensive view of the predictive model's performance in identifying safe or unsafe healthcare data based on various analytical techniques. The models were both trained and tested on the collected dataset.

- **Accuracy**

Accuracy measures the proportion of correct predictions made by the model relative to the total number of predictions. It is calculated using the formula:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

The accuracy score is determined by comparing the true labels (actual values of data points) with the predicted labels (values predicted by the model). High accuracy indicates that the model is effective across the dataset.

- **Precision**

Precision evaluates the model's ability to correctly identify positive predictions. It is calculated as the ratio of true positives to the total number of positive predictions (true positives + false positives):

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

A high precision value indicates that the model minimizes false positives and is effective in identifying positive cases.

- **F1 Score**

The F1 score provides a balanced measure of a model's precision and recall by calculating their harmonic mean. It is particularly useful when there is an imbalance between classes in the dataset. The F1 score is given by:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 score ranges from 0 to 1, with a higher value indicating better performance.

- **Recall**

It measures the proportion of true positive predictions out of all actual positive instances. High recall indicates few false negatives.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **Evaluation approach**

Using the selected combination of features for this study, we assessed the predictive performance of the ML models. Three ML models are evaluated and compared based on the defined metrics. A comparative analysis was conducted to identify the model that best fits each data column for prediction.

VI. Overall Working Principle of the Proposed MQTT and ML-Based Health Monitoring

This section explains the fundamental operating principle of the proposed system. Figure 1 illustrates the architecture of the MQTT and ML-based health monitoring system. The proposed health monitoring system integrates a microcontroller with three primary biomedical sensors, including a body temperature sensor module, pulse oximeter, and HR sensor. The body temperature sensor, DS18B20, is attached to the left arm, while the MAX30100 sensor is positioned on the right index finger. These sensors are connected to the microcontroller through programmed operations, enabling seamless data collection.

The system utilizes Mosquitto, an open-source MQTT broker, with RPI 3 acting as the MQTT broker. When a patient's vital signs exceed predefined threshold values, an alert message is sent to multiple users (doctors or nurses) via the MQTT Dashboard Application on their mobile devices. Sensor data are transmitted to a local or cloud server through both wired and wireless connections. For wired connectivity, an Ethernet port links the microcontroller to the network gateway, enabling data transmission to the server.

The sensors collect data at 2 ms intervals, measuring parameters such as SpO₂ level and pulse rate. The gathered data are displayed locally and stored in a cloud database. Both patients and doctors can access these data via a secure portal. Doctors can

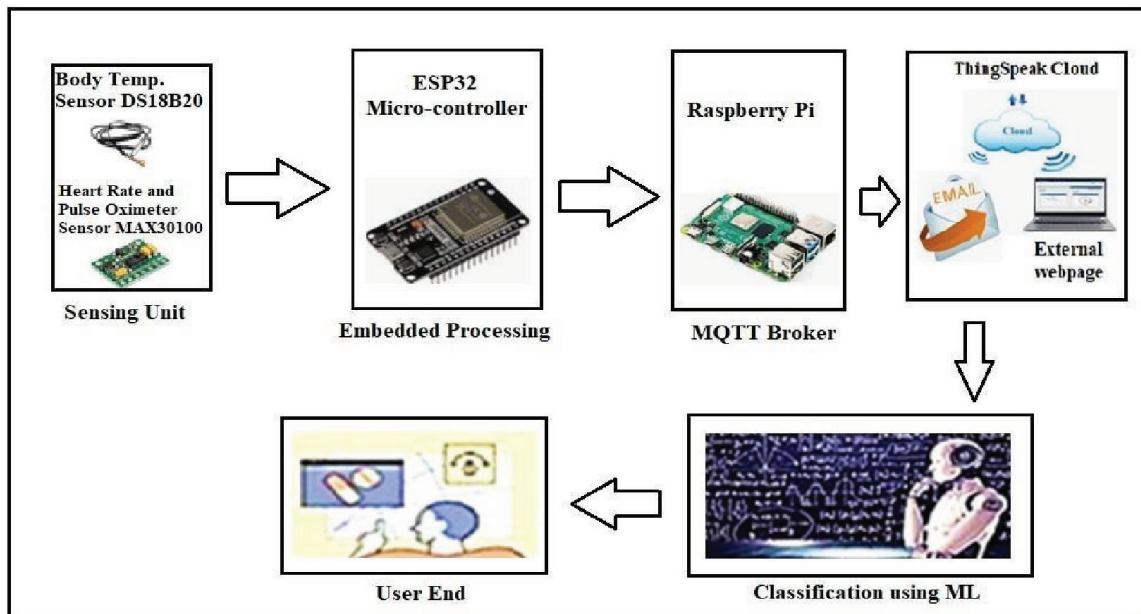


Figure 1: Proposed MQTT and ML-based health monitoring system. ML, machine learning; MQTT, message queuing telemetry transport.

remotely monitor patients and provide guidance. If critical health indicators suggest a serious condition, a notification prompts the medical team to act, including a pop-up option to call for an ambulance.

The system measures essential physiological parameters, such as HR, SpO_2 , and body temperature—and makes the data accessible via an IoT platform. In addition, the system employs ML algorithms to predict a patient's condition even in the absence of remote doctors or internet access. This health indicator unit ensures timely and intelligent health status predictions.

a. Flowchart of classification using ML

- **Import dataset:** Load the dataset into the system.
- **Preprocess data:** Clean the dataset by removing null values.
- **Optimize data:** Transform data columns into a 1D array for classification. Convert string datatypes into integer values using NumPy for compatibility.
- **Select kernel:** Choose a kernel for the classification algorithm (e.g., linear kernel for SVM).
- **Classification:** Apply three different classifiers (KNN, SVM, and DT) to classify columns, such as HR, SpO_2 , and temperature. Measure the model's precision and accuracy.

- **Train model:** Train the classifiers using the dataset.
- **Test system:** Test the trained system by predicting cardiac conditions for input test data. Repeat the process for other classifiers.
- **Comparison:** Graphically compare precision, recall, F1 score, and accuracy across classifiers.

By analyzing HR, body temperature, and SpO_2 levels, this system provides personalized care for chronic conditions. It leverages IoT and cloud technologies for continuous monitoring, with data being processed and analyzed by three classifiers. Comparative analysis identifies the most effective classifier, and the data are sent directly to the medical cloud server.

b. Flowchart of the overall working principle

- **A. Power ON the system**
- **B. Initialize the ESP32 microcontroller**
- **C. Set up the MQTT protocol**
- **D. Select the health parameter to measure**
 - Press push-button 1 to measure HR and SpO_2
 - Press push-button 2 to measure body temperature
- **E. Display the measured parameters on the LCD screen**

- **F. Evaluate the data**
 - If within threshold values, a green LED blinks
 - If outside threshold values, a red LED blinks, and a buzzer sounds
- **G. Data upload**
 - If internet connectivity is available, upload measured data to the cloud server for access by remote doctors.
 - In emergencies, doctors receive an immediate email notification.
- **H. Offline operation:**
 - If the internet is unavailable, the system uses its ML model to predict the patient's condition based on measured data and personal details like age and gender.
 - Predicted results are stored in the cloud for later access.

VII. Experimental Setup, Results, and Performance of ML Algorithms and FPGA-Based Hardware Testing

This section provides an overview of the experimental setup for the proposed health monitoring system, its results, and the FPGA-based hardware testing of ML algorithms and a comparative study of this work with existing previous work.

a. Experimental setup and results

The design and implementation of the proposed architecture are described in this section. Figures 2A–2D illustrate the hardware implementation of the health monitoring system. The system displays

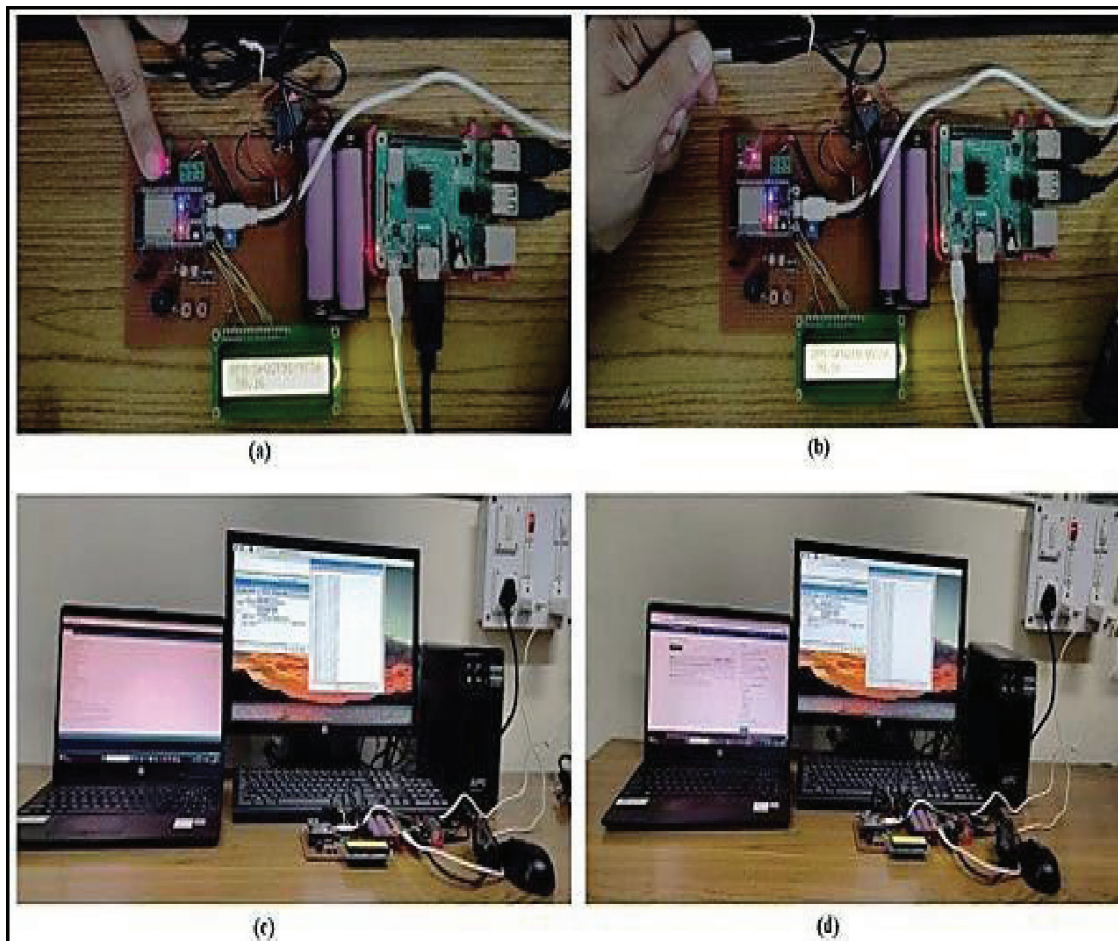


Figure 2: Hardware implementation of the proposed MQTT and ML-based smart health monitoring system (A) SPO2-HR monitoring (B) body temperature monitoring (C, D) overall experimental set up. ML, machine learning; MQTT, message queuing telemetry transport.

temperature readings on the LCD every 3 s, while HR and SpO₂ values are updated every 15 s.

Data can be optionally sent to the ThingSpeak cloud [43]. Table 3 presents vital health parameters as displayed on the LCD and ThingSpeak cloud server during different sessions. The ThingSpeak channel also provides a graphical visualization of the sensor data, making it accessible to doctors and physicians through external webpages. In case of abnormal health parameters, the system alerts the patient via a buzzer and LED. When internet connectivity is available, vital signs such as HR, SpO₂, and body temperature are immediately uploaded to the ThingSpeak cloud server [43].

Figures 3A–3C demonstrate the uploaded data for a single patient and show the graphical representations of vital parameters such as SpO₂ (%), HR (bpm), and temperature (°F) over time for a patient, respectively. Table 3 displays vital health parameters data, both for local visualization and cloud-based monitoring.

b. Performance of ML algorithms and hardware testing using the PYNQ Z2 FPGA board

The performance analysis of three ML classifiers—KNN, DT, and SVM—is carried out by evaluating metrics such as accuracy, precision, recall, and F1-score. Results are summarized in Tables 4 and 5.

KNN demonstrates superior performance across all metrics, particularly in accuracy, precision, recall, and F1-score. By contrast, DT and SVM exhibit lower performance, especially for HR and

SpO₂. Figures 4–7 illustrate the graphical comparisons of accuracy, precision, F1-score, and recall among the classifiers.

c. Hardware testing of ML algorithms using PYNQ Z2 FPGA board

PYNQ (Python Productivity for ZYNQ) combines programmable logic (PL) and programmable system (PS) within the Xilinx ZYNQ FPGA system-on-chip (SoC) [44, 45]. The integration of Python and programming libraries simplifies FPGA-based embedded system development. Figure 8 shows the PYNQ Z2 FPGA board [44]. Using the Vivado Design Suite, we created and connected an AXI Stream interface IP to the PS part of the FPGA block design.

The ML model, developed in Python, is deployed on PYNQ via Jupyter Notebook. The KNN hardware accelerator overlay design, illustrated in Figure 9, enhances the classifier's performance by leveraging PL as a hardware accelerator [45, 46]. Figure 10 outlines the joint simulation system that links Python programs to FPGA hardware through an AXI interface. Figure 11 FPGA presents overall experimental set up for FPGA-based implementation.

Implementation results for KNN and SVM classifiers are shown in Table 6. Table 6 summarizes the hardware implementation results of KNN and SVM classifiers, detailing parameters such as accuracy, clock frequency, resource utilization, and power consumption. Among the classifiers, KNN is more efficient, occupying fewer hardware resources and consuming less power (0.228 W).

Table 3: Vital health parameters displayed locally and on the cloud (ThingSpeak)

Vital parameter	Session	Displayed on LCD	Uploaded to ThingSpeak cloud
Body temperature	Morning	96.35°F, 96.51°F	96.35°F, 96.51°F
	Afternoon	98.24°F, 97.20°F	98.24°F, 97.20°F
	Evening	98.57°F, 96.24°F,	98.57°F, 96.24°F, 97.13°F
HR	Morning	78 bpm, 86 bpm	78 bpm, 86 bpm
	Afternoon	74 bpm, 80 bpm, 76	74 bpm, 80 bpm, 76 bpm
	Evening	68 bpm, 89 bpm, 93	68 bpm, 89 bpm, 93 bpm
SpO ₂	Morning	98%, 97%	98%, 97%
	Afternoon	96%, 81%	96%, 81%
	Evening	95%, 96%	95%, 96%

HR, heart rate; SpO₂, blood oxygen saturation.

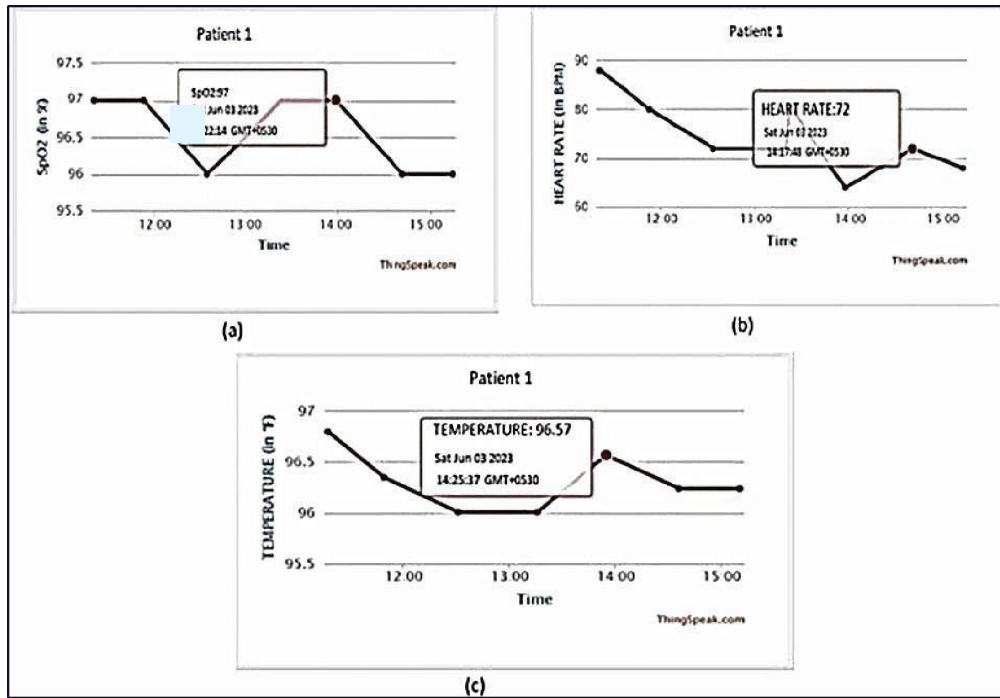


Figure 3: Graphs showing monitoring (A) SpO₂, (B) HR, and (C) body temperature of a patient. HR, heart rate; SpO₂, blood oxygen saturation.

Table 4: Comparison table of accuracy and precision for three different classifiers

Health parameters	Accuracy (%)			Precision		
Classifiers	KNN	DT	SVM	KNN	DT	SVM
HR	98.88	42.78	48.33	0.97	0.35	0.18
SpO ₂	87.22	51.66	58.33	0.87	0.51	0.36
Body temperature	90.55	98.67	95.06	0.91	0.97	0.96

DT, decision tree; HR, heart rate; KNN, K-nearest neighbors; SpO₂, blood oxygen saturation; SVM, support vector machine.

Table 5: Comparison table of F1 score and recall for three different classifiers

Health parameters	F1 score			Recall		
Classifiers	KNN	DT	SVM	KNN	DT	SVM
HR	0.98	0.21	0.20	0.98	0.32	0.36
SpO ₂	0.90	0.39	0.40	0.92	0.50	0.56
Body temperature	0.93	0.92	0.96	0.94	0.93	0.97

DT, decision tree; HR, heart rate; KNN, K-nearest neighbors; SpO₂, blood oxygen saturation; SVM, support vector machine.

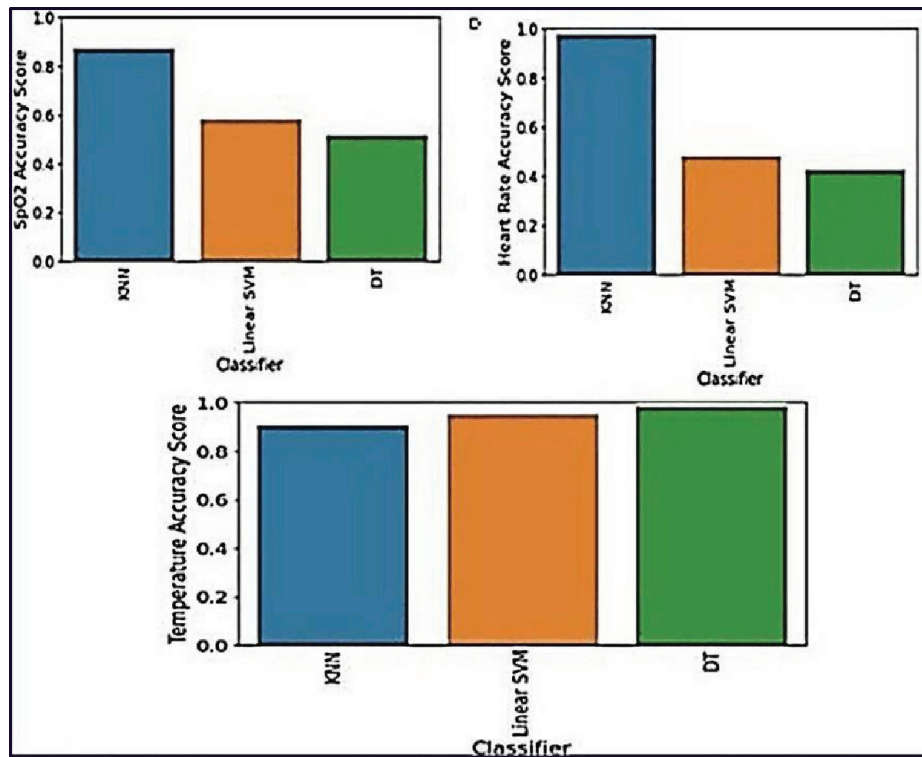


Figure 4: Graphical representation of accuracy for three different classifiers. DT, decision tree; SVM, support vector machine.

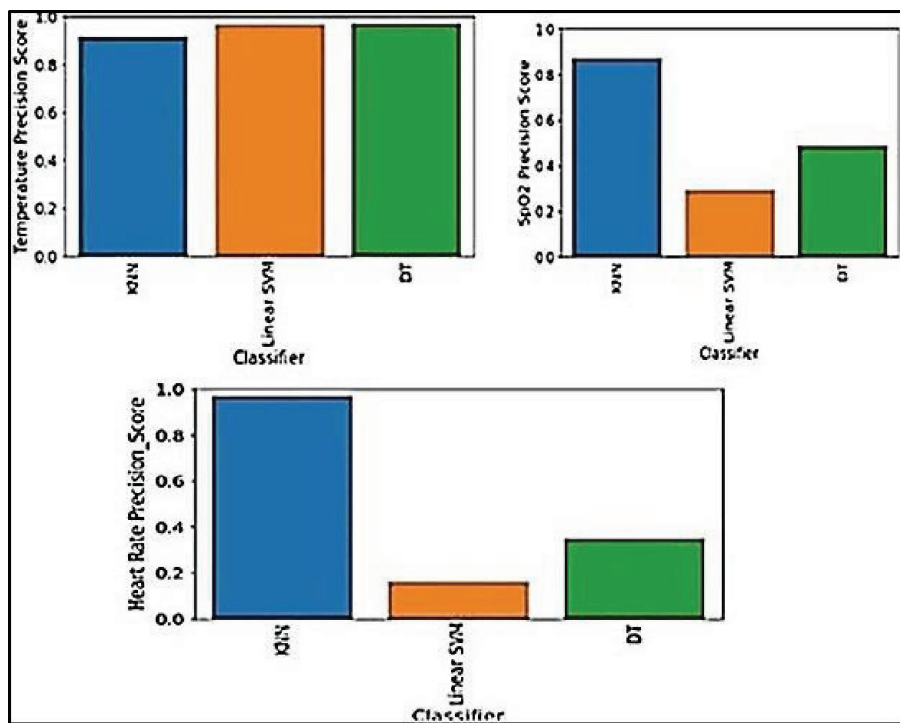


Figure 5: Graphical representation of precision for three different classifiers. DT, decision tree; SVM, support vector machine.

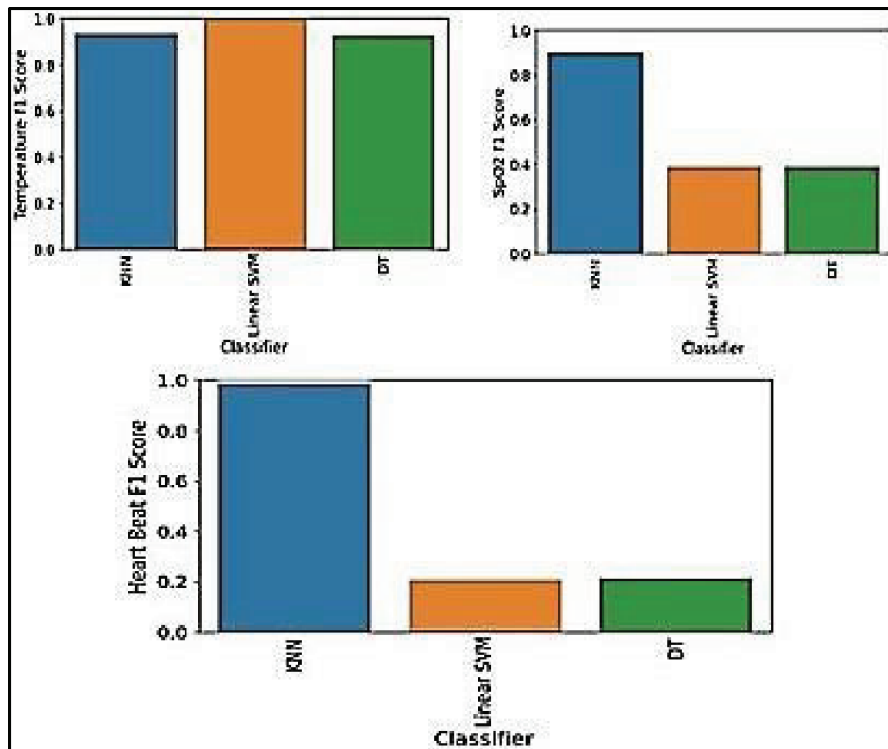


Figure 6: Graphical representation of F1 scores for three different classifiers. DT, decision tree; KNN, K-nearest neighbors.

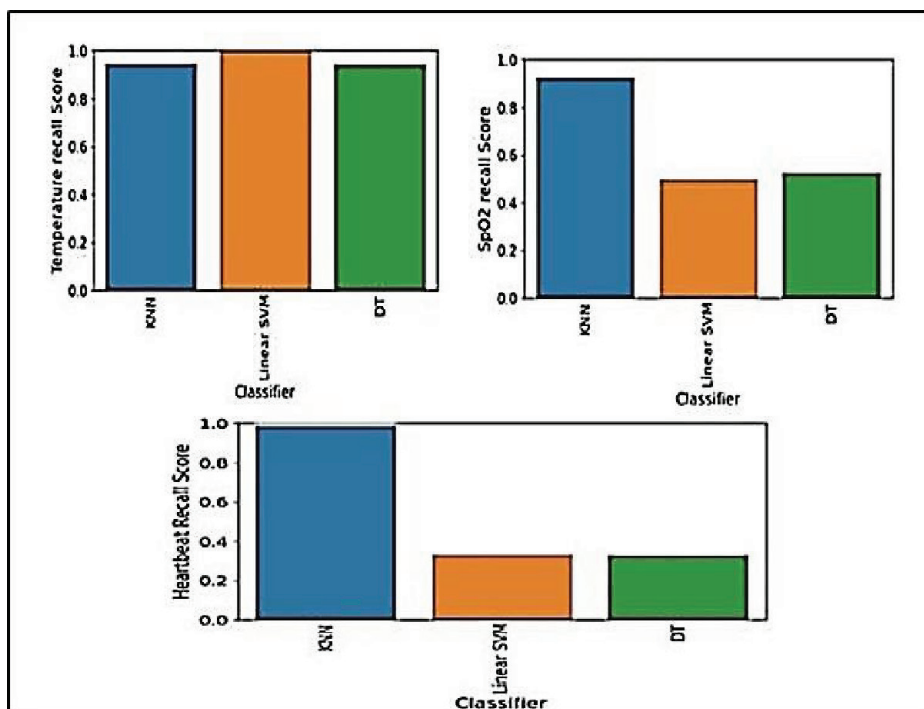


Figure 7: Graphical representation of recall for three different classifiers. DT, decision tree; SVM, support vector machine.

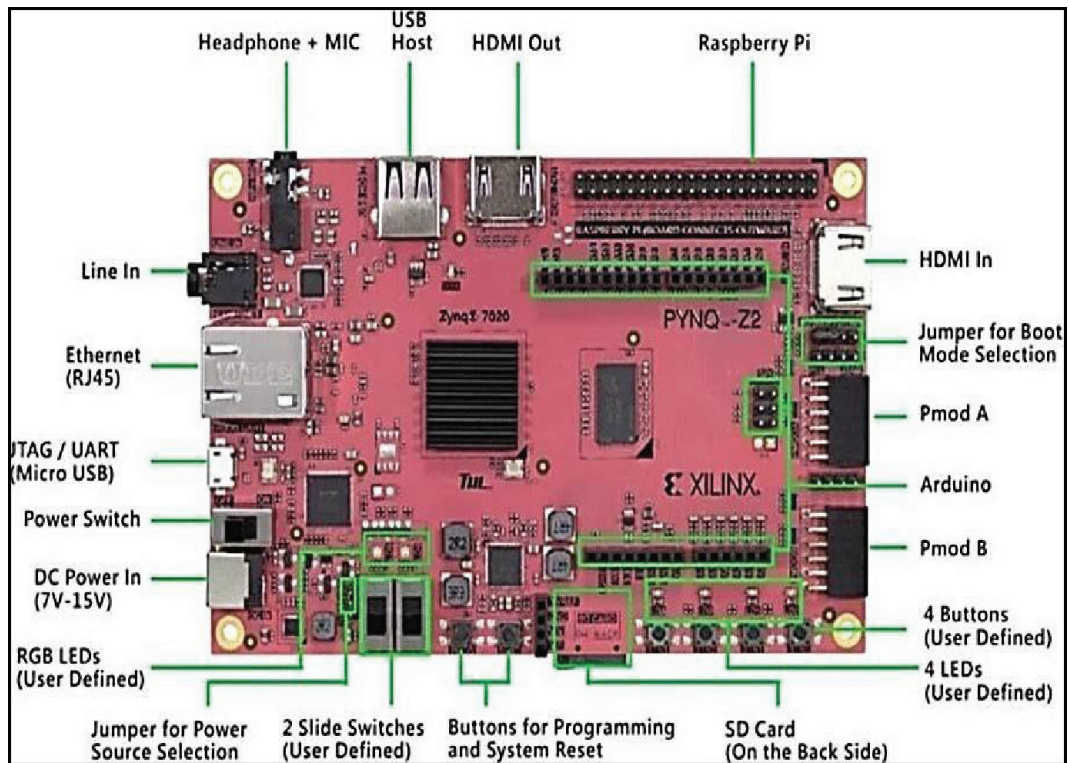


Figure 8: PYNQ Z2 FPGA board.

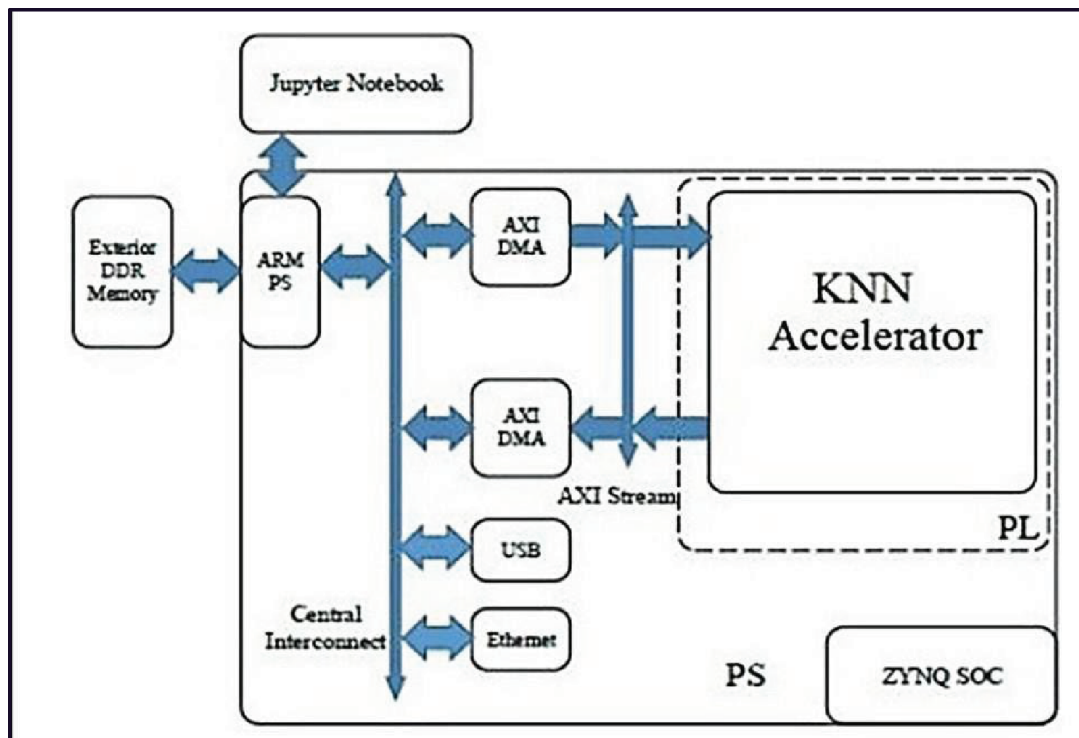


Figure 9: KNN accelerator data path diagram. KNN, K-nearest neighbors; PL, programmable logic; PS, programmable system.

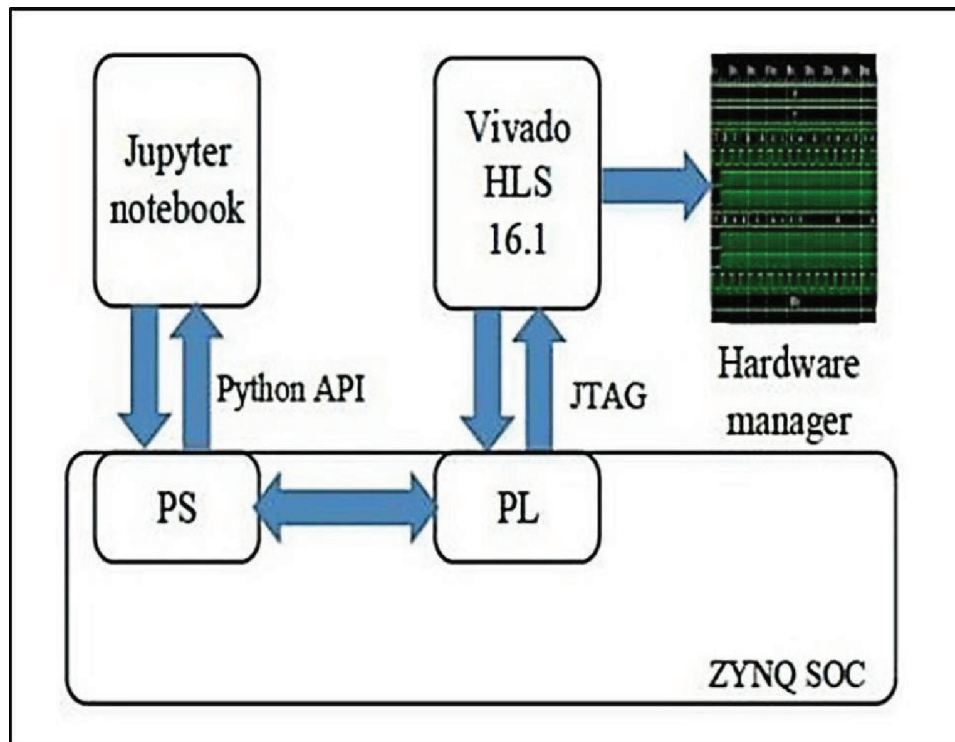


Figure 10: Joint simulation system. PL, programmable logic; PS, programmable system.

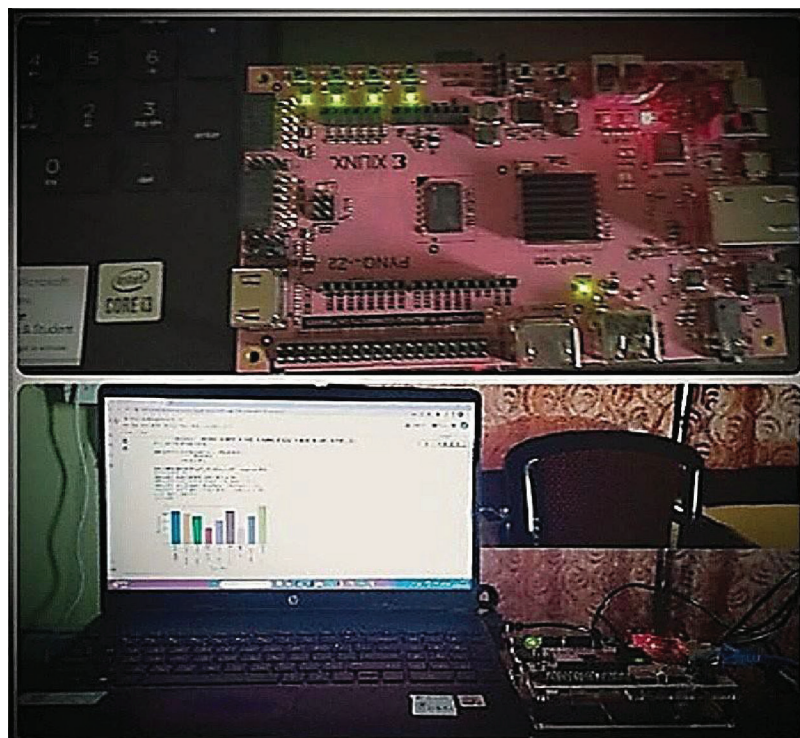


Figure 11: Presents the experimental setup for the FPGA-based implementation.

Table 6: FPGA implementation results for KNN and SVM

Design parameters	KNN	SVM
Classification accuracy (overall in %)	85.55	62.22
Frequency (MHz)	100.07	32.28
BRAM	4	5
DSP48E	10	10
FF	3,493	3,999
LUTs	2,826	3,022
Power consumption (W)	0.228	0.414

FF, flipflop; KNN, K-nearest neighbors; LUTs, look-up tables; SVM, support vector machine.

It achieves an overall classification accuracy of 85.55%, with faster processing times compared to SVM. The results demonstrate that deploying KNN on PYNQ Z2 FPGA accelerates the classification process while maintaining high accuracy and efficient resource utilization.

d. Discussion on KNN's superior performance

The experimental results clearly indicate that the KNN classifier outperforms SVM in terms of both classification accuracy (85.55% vs 62.22%) and hardware efficiency on the PYNQ-Z2 FPGA platform. This superior performance is primarily due to KNN's simple, instance-based learning approach, which avoids complex training phases and extensive parameter tuning, making it particularly effective for the small and structured dataset used in this work. Additionally, KNN's straightforward distance computation and voting mechanism map efficiently onto FPGA resources such as look-up tables (LUTs) and flip-flops, resulting in reduced hardware complexity and a higher operating frequency (100.07 MHz for KNN vs 32.28 MHz for SVM). The integration of lightweight communication protocols further complements KNN's predictable and low-latency operations, leading to lower power consumption (0.228 W compared to 0.414 W for SVM) and higher throughput, which are crucial for real-time edge applications. Moreover, the sensor data features used for health monitoring are well-separated and linearly distinguishable, favoring distance-based methods like KNN. SVM's performance may degrade in such scenarios unless a carefully selected kernel is employed, which was avoided to minimize hardware overhead.

e. Comparative study of proposed work with existing similar works

We used a comparative analysis using tabular format that places our proposed PYNQ-Z2 FPGA implementations of KNN and SVM classifiers side by side with previous FPGA-based works, highlighting classification accuracy, power consumption, frequency, cost, and resource usage shown in Table 7.

e.i. Observations and comparative insights

- In the proposed work, KNN design consumes significantly less power (~0.228 W) compared to Fahad Muslim's implementation of KNN on Virtex-7 (~3.136 W) [48], Melanoma detection work on Zynq-SoC (~1.69 W) [49], Laxmisagar and Hanumantharaju [50] work on Kintex-7 (~4.57 W), and Mhaouch et al. [51] work on PYNQ-Z2 (~1.4 W). Accuracy of KNN classifier is also moderately high 85.55% on proposed PYNQ-Z2 hardware platform shown in Table 7.
- SVM has lower accuracy (62.2%) but also much lower power (~0.414 W) consumes compared to previous works shown in Table 7.
- Resource usage (BRAM, DSP, flipflop [FF], LUT) in the proposed model appears modest and efficient: BRAM 4–5, DSP48E 10, LUT ~2,800 to 3,000, and FF ~3,500 to 4,000.
- The proposed working model implemented using KNN and SVM classifiers on PYNQ-Z2 hardware platform also offers the cost optimized solution with recent existing work shown in Table 7.

Our proposed design shows low power and low resource usage for both KNN and SVM on

Table 7: Comparison of proposed FPGA implementation results for KNN and SVM with existing similar research work

Research work/ hardware (FPGA) platform	Classifiers	Accuracy (%)	Freq. (MHz)	Power (W)	BRAM	DSP's	LUT	FF's	Cost
PYNQ-Z2 (proposed work)	KNN SVM	85.55 62.22	100 32.28	0.228 0.414	4 5	10 10	3,493 3,999	2,826 3,022	Low power/ cost optimized
Virtex-7 implement KNN Algorithm on FPGA [48]	KNN	-		3.136	512	12	23,892	11,838	High cost
ZYNQ-SOC Melanoma detection based on FPGA (2019) [49]	SVM	97.9		1.69	-	-	-	-	~3% of FPGA resource usage and costly
Kintex-7 FPGA implementation on of breast Cancer (2023) [50]	SVM	91.08		4.57	-	-	-	-	High end FPGA. Costly.
PYNQ-Z2 Real-time breast cancer Classification (2025) [51]	CNN	89.87		1.4	~65% of BRAM usage	~72% of DSP usage	~68% of LUT usage	-	High resource usage and moderate cost

FF, flipflop; KNN, K-nearest neighbors; LUT, look-up table; SVM, support vector machine.

PYNQ-Z2 FPGA. It offers moderate accuracy (especially for KNN), but with much lower power consumption than many previous works. So, in summary, we can say that KNN stands out as a low-power (~0.228 W) and moderate-accuracy (~85.55%) inference engine—highly suitable for embedded/edge deployments.

VIII. Conclusion and Future Scope

The proposed health monitoring system offers a versatile solution, suitable for installation at the user's residence or workplace, enabling real-time patient data

exchange with doctors through internet-connected smart devices. Designed to track critical health metrics, the system can also be adapted into a wearable device, showcasing a significant advancement in healthcare technology. By leveraging webhook applets for automated notifications to doctors, the system introduces an innovative approach to patient care. This study highlights the potential of IoT technologies in transforming healthcare by enabling continuous patient monitoring, early detection of abnormalities, and intelligent data analysis through ML algorithms. The system, built on the MQTT protocol, effectively monitors vital parameters, such as HR and body temperature,

offering benefits, such as reduced hospital visits and improved accessibility, even in remote locations. Comprehensive testing of the prediction model was conducted using various ML classifiers, including SVM, KNN, and DT. Among these, the KNN classifier demonstrated superior performance and was successfully implemented using Python on the PYNQ-Z2 FPGA platform through Jupyter Notebook, with a KNN hardware accelerator designed to enhance its functionality. This study represents a step toward the development of intelligent, cost-effective, and efficient healthcare systems capable of improving patient outcomes and healthcare accessibility.

a. Limitations in real-time performance of proposed method

Although the proposed KNN-based health monitoring system on the PYNQ-Z2 platform demonstrates superior classification accuracy and efficient hardware utilization, but there are several limitations affect its real-time performance. First, KNN is a lazy learner, meaning that during inference, the algorithm requires real-time computation of distances between the incoming data and all stored instances. As the dataset size grows, this can lead to increased latency and reduced throughput, particularly in continuous monitoring scenarios. Second, while FPGA resources, such as LUTs and BRAM, are utilized efficiently, the limited on-chip memory of PYNQ-Z2 restricts the ability to handle larger feature sets or high-frequency sensor data streams without external memory support. Additionally, the implemented lightweight protocols, while power-efficient, may introduce network latency when transmitting large volumes of sensor data for real-time analysis. Finally, the model lacks adaptive learning capabilities; any update to the dataset requires reloading or reconfiguring the stored instances, which could hinder dynamic performance in real-world deployments.

b. Future scope

The proposed framework can be further enhanced by integrating wearable and IoT-enabled health devices to enable continuous real-time monitoring and personalized healthcare services. Incorporating advanced ML paradigms, including deep learning technique and ensemble-based hybrid ML models, will significantly improve predictive accuracy and adaptability to diverse health conditions. Moreover, hardware-aware optimizations, such as model

pruning, quantization, and compression techniques, can minimize latency and power consumption for edge deployments. Memory optimization using external DDR memory or advanced memory management techniques to handle high-frequency sensor streams and large feature sets. Finally, secure cloud-edge collaboration and privacy-preserving methods, such as federated learning or blockchain-based data sharing, can be explored to achieve scalable, reliable, and privacy-centric healthcare solutions.

References

- [1] Al-Fuqaha A., et al., "Internet of things: "A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys and Tutorials, vol. 17, no. 4, pp. 2347-2376, 2015.
- [2] Tiwari D, Prasad D, Guleria K, and Ghosh P., "IoT based Smart Healthcare Monitoring Systems: A Review," 2021 6th International Conference on Signal Processing, Computing and Control (ISPC), Solan, India, 2021, pp. 465-469, doi: 10.1109/ISPC53510.2021.9609393.
- [3] Verma Himanshu, Chauhan Naveen, Awasthi Lalit Kumar, "A Comprehensive review of "Internet of Healthcare Things": Networking aspects, technologies, services, applications, challenges, and security concerns, Computer Science Review, Volume50,2023,100591, ISSN 1574-0137, <https://doi.org/10.1016/j.cosrev.2023.100591>.
- [4] Medvediev I., et al., "IoT solutions for health monitoring: analysis and case study," 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), IEEE, 2018.
- [5] Tonello S., et al., "Design and in vitro Characterization of a Wearable Multi-sensing System for Hydration Monitoring," in IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1-11, 2024, doi: 10.1109/TIM.2024.3369161.
- [6] Sharma G, Joshi A. M., Yadav D, and Mohanty S. P., "A Smart Healthcare Framework for Accurate Detection of Schizophrenia Using Multichannel EEG," in IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-9, 2023, Art no. 3523409, doi: 10.1109/TIM.2023.3293544.
- [7] Zhang H, Wang Z, Teng C, Kumar S, Li X, and Min R., "Wearable Cardiorespiratory Sensor for Real-Time Monitoring With Smartphone Integration," in IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1-10, 2024, Art no. 7000710, doi: 10.1109/TIM.2023.3338709.

- [8] Priya A. D., and Sundar S., "Health Monitoring System using IoT," 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, 2019, pp. 1-3, doi: 10.1109/ViTECoN.2019.8899434.
- [9] Pardeshi V., et al. "Health monitoring systems using IoT and Raspberry Pi—a review," 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), IEEE, 2017.
- [10] Divakaran S., et al., "IoT clinic-Internet based patient monitoring and diagnosis system," 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), IEEE, 2017.
- [11] Uddin M. S., Alam J. B., and Banu S., "Real time patient monitoring system based on Internet of Things," 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), IEEE, 2017.
- [12] Bouslama A., Laaziz Y., Tali A., and Eddabbah M., "AWS and IoT for real-time remote medical monitoring," International Journal of Intelligent Enterprise, vol. 6, no. 2-4, pp. 369-381, 2019.
- [13] Stradolini F., et al., "Live demonstration: An IoT smartwatch-based system for intensive care monitoring," 2017 IEEE Biomedical Circuits and Systems Conference (BioCAS), IEEE, 2017.
- [14] Zilani K. A., et al., "R3 HMS, An IoT Based Approach for Patient Health Monitoring," 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), IEEE, 2018.
- [15] Alshammari Hamoud H., "The internet of things healthcare monitoring system based on MQTT protocol". Alexandria Engineering Journal, Volume 69, 2023, Pages 275-287, ISSN 1110-0168, <https://doi.org/10.1016/j.aej.2023.01.065>.
- [16] Maxim Integrated. MAX30100 Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health, Rev 0 (2014), <https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf>
- [17] Kaur A., and Jasuja A., "Health monitoring based on IoT using Raspberry Pi". 2017, International Conference on Computing, Communication and Automation (ICCCA), 1335-1340. doi: 10.1109/CCAA.2017.8230004
- [18] Basu S., Saha S., Pandit S., and Barman S., "Smart health monitoring system for temperature, blood oxygen saturation, and heart rate sensing with embedded processing and transmission using IoT platform," Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019, Springer Singapore, https://doi.org/10.1007/978-981-13-9042-5_8
- [19] Basu S., Ghosh M., and Mandal S. B., "Intelligent remote health monitoring system," International Journal of Information Technology, Communications and Convergence (IJITCC), vol. 3, no. 4, pp. 259-275, <https://doi.org/10.1504/IJITCC.2020.112453>
- [20] Basu S., Ghosh M., and Barman S., "Raspberry PI 3B+ based smart remote health monitoring system using IoT platform," Proceedings of the 2nd International Conference on Communication, Devices and Computing: ICCDC 2019, Springer Singapore, https://doi.org/10.1007/978-981-15-0829-5_46
- [21] Jana J., Basu S., Tripathi S., Chakrabarti A., and Bhaumik J., "Design and Implementation of IoT- Based System for Tracking and Monitoring of Suspected COVID-19 Patient," Advances in VLSI and Embedded Systems: Select Proceedings of AVES 2021, Springer Nature Singapore, doi:10.1007/978-981-19-6780-1_14
- [22] Jana J., Tripathi S., Bhattacharya A., Chowdhury R. S., Karmakar A., Ranjan D., and Bhaumik J., "An Android-Application-Controlled Car for Human Safety Against COVID-19," Internet of Things and Its Applications: Select Proceedings of ICIA 2020, Springer Nature Singapore, 2022.
- [23] Chooruang K., and Mangkalakeeree P., "Wireless Heart Rate Monitoring System Using MQTT," Procedia Computer Science, vol. 86, pp. 160-163, 2016.
- [24] Hussain H. M., Benkrid K., and Seker H., "An adaptive implementation of a dynamically reconfigurable K-nearest neighbour classifier on FPGA," 2012 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), pp. 205-212, June 2012.
- [25] Vieira J., Duarte R. P., and Neto H. C., "knn-stuff: knn streaming unit for fpgas," IEEE Access, vol. 7, pp. 170 864-170 877, 2019.
- [26] Samiee A., Huang Y., and Bai Y., "Frlm: Empowering K-nearest neighbor (knn) through FPGA-based reduced-rank local distance metric," 2018 IEEE International Conference on Big Data (Big Data), pp. 4742-4746, Dec 2018.
- [27] Danopoulos D., Kachris C., and Soudris D., "FPGA acceleration of approximate knn indexing on high-dimensional vectors," 2019 14th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), pp. 59-65, July 2019.
- [28] Wang L-H, Hsiao Y-M, Xie X-Q and Lee S-Y, "An Outdoor Intelligent Healthcare Monitoring Device for the Elderly", IEEE Trans. on Consumer Electronics, vol. 62, no. 2, pp. 128-135, May 2016.
- [29] Bharadwaj H. K., et al., "A Review on the Role of Machine Learning in Enabling IoT Based Healthcare

Applications,” in IEEE Access, vol. 9, pp. 38859-38890, 2021, doi: 10.1109/ACCESS.2021.3059858.

[30] Mullick A., and Sengupta S., “Machine Learning-Based Analysis of IoT Healthcare Data: A Review of Contemporary Research,” 2024 International Conference on Computer, Electrical & Communication Engineering (ICCECE), pp. 1-9, 2024.

[31] Jha S. K., Sagar S., Kumar R., and Narayan U., “Smart Health Monitoring System – A Comparative Study,” 2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), pp. 119-124, 2023.

[32] Gupta P., et al., “IoT based smart healthcare kit,” 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), IEEE, 2016.

[33] Vimal S. P., Vadivel M., Baskar V. V., Sivakumar V. G., and Srinivasan C., “Integrating IoT and Machine Learning for Real-Time Patient Health Monitoring with Sensor Networks,” 2023 4th International Conference on Smart Electronics and Communication (ICOSEC), pp. 574-578, 2023.

[34] Soni Dipa, and Makwana Ashwin., “A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT)”. (ICTPACT – April, 2017).

[35] Bhuvana J., Kumar S., Deepa M., Pavithra G., and Anandkumar V., “AI Innovations in IoT and Machine Learning for Health Prediction Systems,” 2023 International Conference on Inventive Computation Technologies (ICICT), pp. 1432-1435, 2023.

[36] Orpa Tasnim, Ahnaf Adil, Ovi Tareque, and Rizu Mubdiul, “An IoT Based Intelligent Healthcare Solution With ESP32 Using Machine Learning Model”. (2023). 10.1109/STI56238.2022.10103231

[37] Maxim Integrated.: DS18B20 Programmable Resolution 1-Wire Digital Thermometer, Rev4(2015), <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

[38] Kumar N, Sharma A, and Singh R, “K-Nearest Neighbor Based Early Warning System for Health Monitoring,” IEEE Transactions on Information Technology in Biomedicine, vol. 24, no. 3, pp. 519-528, May2020. DOI: 10.1109/TITB.2020.2955478

[39] Zhang X., Zhao Y., and Li M., “Support Vector Machine-Based Classification of Heart Disease Using Wearable Sensor Data,” IEEE Access, vol. 9, pp. 78465-78475, Jul. 2021. DOI: 10.1109/ACCESS.2021.3085029

[40] Wang L., Chen Z., and Zhang H., “Decision Tree Approach for Diabetes Prediction in Real-Time

Health Monitoring Systems,” IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 4, pp. 1248-1257, Apr. 2021. DOI: 10.1109/JBHI.2021.3051115

[41] Singh A., and Gupta P. K., “Hybrid Machine Learning Models Using KNN, SVM, and Decision Tree for Wearable Health Monitoring Systems,” IEEE Sensors Journal, vol. 22, no. 5, pp. 4843-4850, Mar. 2022. DOI: 10.1109/JSEN.2022.3142256

[42] Sokolova S., and Lapalme G., “A Systematic Analysis of Performance Measures for Classification Tasks,” Information Processing & Management, vol. 45, no. 4, pp. 427-437, Jul. 2009. DOI: 10.1016/j.ipm.2009.03.002

[43] Thingspeak.: IoT Analytics. Accessed on 15.10.2024. <https://thingspeak.com>

[44] PYNQ-Z2 FPGA Board. Datasheet Accessed on 10.11.2024. <https://www.amd.com/en/corporate/university-program/aup-boards/pynq-z2.html#specs>

[45] Zhang X., Xiong Y., and Lin T., “Accelerated Neural Network Inference for Biomedical Signal Processing Using PYNQ-Z2 FPGA,” IEEE Access, vol.9, pp.12345-12357,2021 DOI:10.1109/ACCESS.2021.3059876

[46] Xiang Y., and Oniga S., “FPGA-Based Hardware Accelerator for EEG Signal Patterns Recognition,” Sensors, vol. 22, no. 5, pp. 1-16, 2024. DOI: 10.3390/s22051475

[47] Tallapaneni N. S., Jayanthi M. and Venkatesan M., “IoT based Smart Health care system monitoring using Machine Learning Techniques,” 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, 2021, pp. 1-6, doi: 10.1109/ICSCAN53069.2021.952644

[48] K-Nearest Neighbor algorithm: <https://github.com/fahadmuslim/KNN>; Accessed on 21.07.2025

[49] Afifi S., Hosseini H. G., Sinha R., “A system on chip for melanoma detection using FPGA-based SVM classifier”, Microprocessors and Microsystems, Volume 65, 2019, Pages 57-68, ISSN 0141-9331, <https://doi.org/10.1016/j.micpro.2018.12.005>

[50] Laxmisagar H.S., Hanumantharaju M.C, “FPGA implementation of breast cancer detection using SVM linear classifier”. Multimed Tools Appl 82, 41105–41128 (2023). <https://doi.org/10.1007/s11042-023-15121-6>

[51] Mhaouch A., Gtifa W., Machhout M., “FPGA Hardware Acceleration of AI Models for Real-Time Breast Cancer Classification”. AI. 2025; 6(4):76. <https://doi.org/10.3390/ai6040076>