International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I1P102

Volume 73 Issue 1, 14-46, January 2025
© 2025 Seventh Sense Research Group®

Original Article

IoT-Enabled Smart Health Monitoring System with Deep Learning Models for Anomaly Detection and Predictive Health Risk Analytics Integrated with LoRa Technology

D. Antony Pradeesh[1], N. P. Subiramaniyam[2]

[1,2]*Department of Electronics and Communication Systems, Nehru Arts and Science College, Tamil Nadu, India.*

[1]*Corresponding Author* : *antony.pradeesh@gmail.com*

Received: 24 July 2024 Revised: 16 December 2024 Accepted: 23 December 2024 Published: 31 January 2025

Abstract *- The incorporation of Internet of Things (IoT) technology into advanced deep learning models has led to the development of complex health monitoring systems capable of determining anomalies and predicting health risk conditions in real-time. This research illustrates a system driven by Blood Pressure (BP), Heart Rate (HR), oxygen saturation, body temperature, Galvanic Skin Response (GSR), ECG, EMG, and particulate matter, among other sensors, in this study—all coordinated through a Raspberry Pi 5. This research considered three leading models of anomaly detection that exhibit high accuracy in handling diversified health data: Bidirectional Long Short-Term Memory (LSTM) using K-fold Cross Validation, eXtreme Gradient Boosting (XGBoost), and Random Forest. The system uses LSTM and Gated Recurrent Unit (GRU) to predict health risk conditions for health management, including hypertension, hypoxia, cardiac stress, fever, and stress. In the empirical approach, the system indicates impressive precision and accuracy in detecting anomalies and health risk prediction; thus, the system could be improved in remote health monitoring and patient care. Furthermore, the adaptive Long Range (LoRa) communication system ensures reliable data transmission without an internet connection, ensuring that data is transferred only when anomalies are detected. This paper represents the potential of integrating IoT and deep learning in bringing transformational changes in healthcare by providing scalable and efficient solutions for continuous health monitoring with early interventions.*

Keywords *- Smart health care, LoRA communication, Remote patient monitoring, Prediction, IoT.*

1. Introduction

The contemporary unification of various technologies within the Internet of Things (IoT) sphere and deep learning has revolutionized several fields, most importantly healthcare [1], through their functionalities in the process of sophisticated monitoring and real-time predictive analysis of health-related data [2]. The invention of the wearable or embedded sensors made it feasible to continuously monitor health-related metrics like systolic blood pressure (BP_sys) and diastolic blood pressure (BP_dia), pulse rate (HR) and Blood oxygen saturation (Sp02)—important parameters for early diagnosis and preventive health strategies [3]. However, this data transmission and processing magnitude challenges scalability, energy efficiency, and real-time analysis. Based on LoRa technology [4], this proposed research is focused on implementing a new type of IoT health monitoring system to improve data communication, with support for advanced deep learning models to get better predictive accuracy and overall operational efficiency. It has a dual nature in the sense that, at the forefront of this system, there is communication and computation. Recent advancements in Internet of Things (IoT) technology and deep learning have revolutionized healthcare, offering transformative opportunities for continuous health monitoring and predictive analytics. Such breakthroughs can significantly improve patient treatment by facilitating immediate recognition and handling of health irregularities as they occur. However, existing health monitoring systems still face several critical limitations that hinder their widespread adoption and effectiveness. One of the primary challenges is scalability and resource efficiency. Many current systems rely heavily on constant internet connectivity, making them impractical for deployment in resource-constrained or rural environments where connectivity is limited or unreliable. Additionally, many systems can monitor health parameters but lack advanced predictive capabilities. This deficiency prevents the early detection of critical conditions, limiting the systems' ability to provide proactive healthcare solutions. Another significant limitation is the fragmented integration of multivariate sensor data. Most existing systems fail to efficiently aggregate and analyze data from multiple sensors, leading to incomplete patient health assessments. This gap in sensor integration often results in reduced accuracy and the

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

overall efficiency of patient monitoring infrastructures. By incorporating adaptive LoRa communication, the system guarantees the continuous and reliable transfer of vital health information to long ranges with certainty, especially in low-resource settings that lack robust infrastructure [5]. This adaptive approach optimizes power consumption, adjusts data transmission constructed on the criticalness of health data, and, therefore, enhances the system's responsiveness and energy efficiency. The latter system utilizes advanced data analytics to understand health metrics in real-time, enabling prompt interventions. Therefore, with strong communication and intense computation, the system becomes very effective for remote health monitoring. Computationally, the system triggers a sequence of deep learning models to undertake real-time analysis over incoming data streams — some of which include deep learning models like LSTM, XGBoost and Gated Recurrent Unit (GRU) networks. The models can be trained to detect abnormal patterns that may show potential health problems in a patient and thus develop ways of intervening on time. These models are woven into the IoT architecture, which, in real-time, continuously monitors health seamlessly and automatically; otherwise, it burdens the medical personnel while immediately giving feedback and guidance to patients [6].

1.1. Problem Statement

Though there have been advancements in analytics and data processing technology, modern health monitoring is far from perfect. Integrating and analyzing disparate sensor data poses difficulties, making it impossible to create an all-inclusive interpretation of patients' health based solely on this information. Remote Patient Monitoring (RPM) systems play a significant role in eldercare management and chronic disease care [7]. Smart healthcare has now become an indispensable element of healthcare delivery. Prognostication and risk stratification are central but inefficient components of existing health monitoring frameworks, which primarily act in a reactive mode rather than being predictive. An aging global population, combined with delayed onset and limited continuity of care, presents health monitoring systems with few options for proactive interventions to address the near exponential rise of chronic diseases [8]. This requires early detection, continuous monitoring, and preemptive action to mitigate the strain on healthcare systems worldwide [9].

1.2. Motivation

The exponentially growing burden of various chronic diseases worldwide corresponds to a growing requirement for real-time health monitoring. Traditional healthcare systems, however, have huge limitations, which are primarily periodic in health checks and reactive in medical interventions. Integrating emerging technologies from the Internet of Things with advanced deep-learning techniques presents an unprecedented chance to change how healthcare is delivered; this would enable continuous, non-invasive monitoring of health conditions in real-time. This paradigm shift is being compelled by early detection of health risks, timely medical intervention and better clinical outcomes. Incorporated as part of such systems, effective processing and transmission of large amounts of sensor data, trustworthiness and secureness of information, and sophisticated algorithm creation are important to drawing meaning from complex data segments.

1.3. Objectives

Therefore, This research is timely, considering the need to upgrade the management of chronic diseases and real-time health monitoring, as well as some of the limitations of traditional healthcare systems by integrating IoT technologies with advanced deep learning. The primary objective is to design a smart system powered by the Internet of Things (IoT) technology health-monitoring device that integrates all health monitoring sensors and utilizes LoRa technology for reliable, adaptive data transmission even in resource-constrained settings. The system aims to dynamically adjust data transmission based on the criticality and network conditions to optimize energy consumption and ensure data reliability. Additionally, implementing and Optimizing deep learning models, specifically LSTM, XGBoost, and GRU networks, supports interpreting continuously generated medical information to prompt detection of abnormal patterns and potential health issues. Comprehensive system performance evaluations in real-world environments assess accuracy, efficiency, and scalability, intending to enhance patient engagement and outcomes by providing immediate health insights and fostering proactive health management. This approach is intended to revolutionize patient care by integrating cutting-edge technology to facilitate a more proactive, personalized, and preventive healthcare paradigm.

1.4. Foundational Justification for Core Research Concerns

The foundational justification of the core research concerns of this study is founded on the critical need to advance healthcare delivery systems using innovative technologies. Traditional health monitoring mechanisms are usually episodic and reactive, which are mostly inadequate for the timely detection and management of chronic conditions requiring continuous observation and swift response. The central research concerns of this study are based on the critical need for reforms in healthcare delivery systems using innovative technologies. Traditional health monitoring mechanisms, usually episodic and reactive, make it difficult to identify the early manifestation and manage chronic conditions requiring continuous observation and rapid response. It is also driven by several pivotal concerns that arise to mitigate these challenges. Increasing pressure from chronic diseases requires more sophisticated monitoring solutions to support continuous real-time data needed in managing and treating those diseases. Integrating IoT devices and sensors in health monitoring helps address this need and opens a wider spectrum of healthcare applications, from preventive measures to acute care interventions. Research says emergency visits and hospitalization could be drastically reduced if not for the case of continuous monitoring that allows earlier intervention measures [10]. The effectiveness of any health monitoring system rests squarely on how well a system can handle and transmit large volumes of data reliably. This can be scaled up in handling these data volumes using LoRa technology through long-range transmission with low power consumption, based on its deployment across varied geographical landscapes, including underserved areas. This is supported by the technology's capability in other sectors to communicate effectively over extensive distances without substantial energy costs.

The complexity of physiologic data thus conduces to a need for robust analytical tools, which would be helpful in the extraction of actionable insights. Deep learning is a powerful framework for processing and analyzing large data sets on their latent features and patterns that are not so obvious to human observers. This research uses LSTM, XGBoost, and GRU models, which have recorded an ace in sequence prediction problems, particularly in recognising temporal anomalies in time series data—a feature critical to conditions like arrhythmias or a sudden drop in Blood Oxygen Levels. The proposed system reduces the hassle for healthcare providers due to automation since it increases operational efficiency in data collection and analysis, necessitating robust analytical tools capable of extracting actionable insights. Deep learning offers a robust platform for handling extensive datasets and uncovering patterns that may remain elusive to human analysts. The inclusion of LSTM, XGBoost, and GRU algorithms in this study is warranted by their proven effectiveness in sequence-based forecasting tasks, especially for detecting time-dependent irregularities, an essential aspect when dealing with conditions such as cardiac strain or abrupt reductions in blood pressure.

1.5. Organization

The subject of this research paper is deploying an innovative smart healthcare monitoring system. Section II designated the literature review, has a comprehensive survey of the existing academic and industry work related to smart healthcare monitoring systems. It surveys relevant literature on existing systems, prior studies, their methodologies, and findings. Finally, this section also sets the stage for the contribution by highlighting the gaps or limitations in the current landscape that the proposed work will overcome. The details of the sensor setup and communication protocol is elaborated in section III, followed by the presentation of the deep learning algorithms. Section IV outlines the results and discussion, providing valuable insights into the proposed work. Further elaboration on the challenges, the proposed and implemented solutions, and potential future work in smart healthcare is presented in the conclusion section.

2. Literature Review

The convergence of the IoT method with deep-learning algorithms has transformed the area of health monitoring into novel opportunities, which in turn further progressed patient care [11]. Modern health-monitoring systems incorporate a significant number of sensors that can gather vital health data continuously to monitor the physiological parameters of the patients, for example, the rate of the heart (HR), blood pressure (BP), and oxygen-saturation levels (SpO2) [12].

These systems enable both immediate medical intervention and predict health care [13]. It can predict potential health issues using advanced analysis before it become critical [14]. A widely adopted approach Deep learning, recognized as a specialized branch within the broader domain of machine learning (ML), which enhances the accuracy of health monitoring systems. These deep learning models analyze extensive datasets from IoT devices, enabling the identification of subtle patterns that might indicate the beginning of the decline in health status [15]. This is particularly important for conditions with constant monitoring, such as heart disease and diabetes [16], since early detection can considerably change the outcomes of treatments and improve patients' quality of life [17]. The IoT-enabled health systems have proved to be easily scalable and flexible enough to find their place in any setting, from urban healthcare facilities to remote areas with a very low level of medical infrastructure [18]. Cloud technologies enable such systems to provide services by working with minimal human oversight to store and process an individual's health data [19].

On the one hand, this ensures that critical health data is always available to health experts and from any location and improves the overall efficiency of health services by reducing dependency on physical healthcare infrastructures [20]. A four-module IoT architecture integrating data acquisition with context-aware computations, achieving high accuracy, scalability, and response times using multiple deep learning algorithms [21], has been developed for real-time smart health care. Another research has uncovered the interoperability between AI and IoT in health monitoring, achieving good accuracy in disease prediction using the Random Forest classifier [22].

An IoT-assisted intelligent monitoring model, integrating deep learning networks with Bayes theorem, achieved an 87.87% accuracy rate in disease prediction, highlighting its relevance. Experimental results show that machine learning models like NB, RF, LR, and Support Vector Machines are good at monitoring and diagnosing diseases such as cardiovascular and diabetes [23]. Research on heart disease prediction using SVM and LR on the Cleveland HD dataset achieved accuracies of 92.37% and 88.67%, respectively [24], demonstrating the feasibility of machine learning for early warning symptoms of heart disease. The Raspberry Pi 4B microcontroller is deployed as an Internet of Things-based concept [25]. The system measures key health indicators, and the sensors used are the DS18B20 and MAX30100. This solution uses the SIM7600E GSM and GNSS HAT module to have real-time data transfer into cloud storage with the capabilities of accessing information regarding the patient on demand by healthcare providers. Most of the research only conduces to underline that successful application of IoT devices is going to change drastically how healthcare would be practiced in terms of improved accuracy of data and dynamism in response to patients' requirements he integration of IoT devices alongside LoRa-based communication channels facilitates continuous, real-time surveillance of patients' health parameters and predictive modelling of cardiovascular parameters using Machine Learning algorithms like ANN, Naïve Bayes, CNN, and LSTM has been developed to mitigate chronic diseases [26]. Embedded systems provide a public health response by making quick, accurate, and reliable data related to the environment available to citizens. The findings underscore the promise of IoT for public health surveillance and the benefit of intelligent systems in embedding in everyday health management practices.

The Internet of Things in health hit a high and advanced level; therefore, researchers focus on the various aspects of this technology to develop and enhance patient monitoring and care. In this regard, the first stage of the research is a cloud-based healthcare system for monitoring emotional stress and anxiety on a real-time basis and sensors coupled with cloud technology [27]. The system used galvanic skin response to measure emotional responses alongside more traditional health metrics internalized to provide some reliability in matching physiological responses to affective states.A new anomaly detection approach has been developed for the Internet of Medical Things; the 3D hybrid model combines ARIMA and decision trees. The combination integrates the strong points of both time series forecasting and decision tree classification for improved accuracy in health anomaly detection. Their initial results show that this approach may drastically improve early detection capacity in IoMT systems, thereby opening the door for trustable outcomes in health care [28].

Table 1. Comparison of existing systems

Study Reference	Methodology	Algorithms Used	Results
AI and IoT [21]	Comparison of ML algorithms for disease prediction in healthcare	Random Forest, Decision Tree, SVM, Naïve Bayes, AdaBoost, ANN, KNN	97.62% accuracy with Random Forest in disease prediction.
IoT-Based Monitoring [22]	IoT sensors in a four-module architecture	BPNN with Adaptive Grasshopper Optimization	83% Accuracy in health monitoring.
Swarm-ANN [23]	Heart Disease Prediction System with two-phase weight modification	Swarm-Artificial Neural Network (Swarm-ANN)	95.78% accuracy in heart disease prediction.
EDLN-BT [24]	IoT-Assisted Health Monitoring with Enhanced Deep Learning Network	Enhanced Deep Learning Network using Bayes Theorem	94.2% Accuracy in health monitoring.
Remote Monitoring [25]	IoT-based remote monitoring of vital health signs	Not specified	Enhanced health monitoring precision and speed using Raspberry Pi 4B and cloud transmission.
Machine Learning models with LoRa Communication [26]	IoT and LoRa communication with AI algorithms	CNN, ANN, Naive Bayes, and LSTM	ANN model showed higher performance with the lowermost Mean Absolute Error (MAE).
IoT based Anxiety and Stress Monitoring [27]	IoT-based healthcare systems focus on emotional and heart rate monitoring	Not specified	Accurate measurements of heart rate, body temperature, SpO2, and emotional levels with GSR sensor, showing 83.3% alignment with subjective emotional assessments.
Anomaly detection [28]	IoT-based anomaly detection using hybrid ARIMA and decision tree models	ARIMA, Decision Tree	Improved accuracy and efficiency in anomaly detection within IoMT systems.
Anomaly detection [29]	IoT-connected sensor-based system for real-time health assessment and anomaly detection	Support Vector Machine (SVM)	Achieved impressive accuracy rates with the SVM model.
COVID Monitoring [33]	LoRa-based COVID patient health detection system	Not specified	Utilizes LoRa for communication in areas with poor internet connection.

An affordable IoT-enabled health monitoring system monitors temperature, blood pressure (BP), and ECG with multi-sensors for assessing patient status in real-time evaluation [29]. It achieved an overall accuracy of 92% and provided test results with an F1 score of 90% using a deep learning model, thus proving that efficient machine learning techniques can forecast health more accurately and quickly. Another highly advanced health monitoring system driven by IoT technology [30] incorporates these environmental factors, such as indoor air quality, which could be significant in respiratory disease. The system combines PM2.5 and PM10 sensor readings for air quality with health metrics, demonstrating the future potential of IoT systems to provide comprehensive data regarding health effects from both physiological and environmental sources.

Therefore, This research paper shows the demand for systems capable of context-aware health monitoring. Another research with the implementation of a system for healthcare monitoring enabled with IoT cloud that predicts and gives alertness to one's medical conditions through smart devices. The systems involve systems in which real-time data comes from sensors working to process vital signs of health parameters, which is accomplished using AI algorithms to predict current and future health conditions [31]. A study points out the usage of Low Power Wide Area Networks for monitoring health, emphasizing the machine learning classifiers that become the primary source of fall detection in a LoRa communication network [32]. The most astonishing result of this system was the accuracy rate of the decision tree classifier, 99.864%, which proves that for real-time health monitoring and intervention, IoT needs to be skilfully amalgamated with machine learning techniques.

Furthermore, another proposed IoT- and LoRa-based system for COVID detection and for monitoring distant areas with poor internet connectivity to monitor the patients [33]. The system is equipped with several sensors for health monitoring and LPWAN technology for reliable data transmission to control the pandemic effectively. Another recently developed energy-efficient IoT system allows continuous remote monitoring of the patients' vital health parameters. Recent research has explored the energy efficient technology for remote patient monitoring systems to analyse patients' health in resource-constrained environments [34]. In recent studies [35]-[39], bio-inspired optimization techniques are now transforming IoT applications in smart healthcare frameworks. These techniques have been shown to significantly enhance network efficiency and security, which are crucial for the secure dissemination of healthcare data.

2.1. Gaps Identified in Literature Review

However, most current research on anomaly detection and predictive analysis of future risks in a health monitoring system driven by IoT technology has a depth and breadth limitation. Instead, Most systems have employed conventional algorithms that might not capture complex patterns in multivariate health data. The models are normally weak in detecting subtle anomalies in health that might point toward severe conditions. Further, though highly efficient in real-time monitoring, most systems often lack high-level predictive modeling, which can accurately predict severe health risks and conditions like hypertension and cardiac stress before manifesting them clinically. While previous studies have successfully used IoT-based solutions integrated with machine learning models, they often employ basic algorithms like SVM, Decision Trees, and Naïve Bayes, which might not capture the temporal patterns present in physiological data. Additionally, most systems are not equipped with adaptive communication mechanisms like LoRa for reliable long-range data transmission. Moreover, most existing systems, such as LoRa, lack adaptive communication mechanisms that facilitate reliable long-range data transmission. Without such mechanisms, these systems face significant challenges in maintaining stable communication over extended distances, particularly in remote or underserved areas. Addressing these gaps is essential for developing next-generation health monitoring systems that are scalable, resource-efficient, and capable of delivering advanced predictive analytics.

Given such challenges, the research proposes a new integration and uses Bidirectional LSTM, XGBoost, and Random Forest algorithms to increase anomaly detection and predictive accuracy within health monitoring systems. Bidirectional LSTM potentially learns the data sequences in both directions, which offers a detailed analysis of trends from temporal data. This is supplemented by the strong decision-making capabilities of XGBoost and Random Forest, which have a layered approach toward detecting and validating health anomalies. In predictive analytics, combining LSTM and GRU models ensures an unprecedented forecasting ability for imminent health risks based on continuous data streams, allowing early intervention strategies for conditions such as hypoxia and cardiac stress. Thus, these high-end methodologies make the health monitoring system reactive and proactive, changing the face of patient care through early detection and timely medical responses.

3. Methodology

The methodology segment of this scholarly article is designed to address different technological and analytical approaches that can be used in IoT-based health monitoring systems. It has seven major divisions: IoT-based Real-time Health Monitoring System, detailing multiple sensor integrations with a central role to the Raspberry Pi 5, and LoRa-based data transmission, explaining adaptive LoRa communication for effective data transmission in remote areas. Three models are utilized for anomaly detection: Bidirectional LSTM with K-Fold Cross-Validation (BiLSTM-KFCV), XGBoost, and Random Forest. Two models are used for health risk prediction: LSTM and GRU (Gated Recurrent Unit).

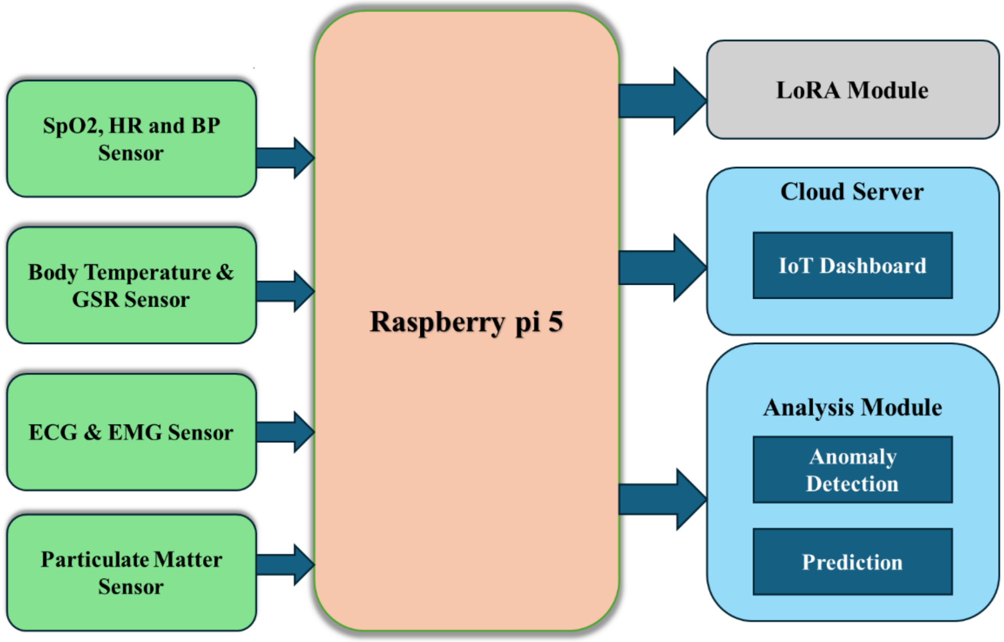

Fig. 1 Block diagram

3.1. IoT-Based Real-Time Health Monitoring System

Healthcare represents a primary domain where embedding IoT technology into systems that track vital health metrics has developed to such a state that it would be feasible for critical health parameters to be continuously monitored with real-time accuracy. The information captured from various physiological sensors can be transmitted and displayed through this advanced system, which has its abilities modeled on IoT. It helps the health provider and patient access important health metrics effectively by attaching these sensors to an IoT dashboard through a Raspberry Pi 5.

These sensors monitor and measure the relevant physiological parameters: blood oxygen saturation, parameters such as Heart Rate (HR), Galvanic Skin Response (GSR), and both systolic and diastolic blood pressure readings. Other physiological parameters include body temperature, an electrocardiogram, electromyography, and particulate matter levels. The Raspberry Pi 5 is a data aggregator from these sensors and sends this information to the cloud for later processing and visualization. The sensors are well-placed to obtain accurate continuous readings; each has a different functionality.

The SpO2 sensor measures blood oxygen saturation, which is important in diagnosing hypoxia and other illnesses. Therefore, the heart rate sensor would help detect possible arrhythmia or other conditions related to the cardiac system. Skin electrical conductance measured by the GSR sensor allows assessment of the activity of sweat glands, which alter depending on the emotions. Blood pressure sensors monitor arterial pressure during the contraction and relaxation phases of the heartbeat, thus helping identify hypertension. The body temperature sensors monitor the body's temperature to indicate fever or infections. All these sensors are interfaced with the Raspberry pi 5 processor, as depicted in Figure 1.

The ECG Sensor measures the heart's electrical activities, providing information regarding cardiac health. Muscle activity can be determined through the EMG sensor, thus helping diagnose neuromuscular disorders. Finally, particulate matter sensors such as PM1, PM2.5, and PM10 measure air quality, vital for respiratory health. This data is continuously collected to keep the system updated.

3.2. Data Transmission to the Cloud

The Raspberry Pi 5 is pivotal in relaying sensor-derived information to cloud-based resources. It connects to Wi-Fi and thus should ensure stable and swift data transmission. Upon reaching the cloud, the data is stored and visualized in real-time through an IoT dashboard. The process of transmitting data from the Raspberry Pi to the cloud can be broken down into several stages.

3.2.1. Data Collection

$$Dcollected = \sum_{i=1}^{n} S_i(t) \tag{1}$$

Where $Dcollected$ represents the total data collected from all sensors (S_i) at the time (t).

3.2.2. Data Preprocessing

$$D_{\text{preprocessed}} = f_{\text{preprocess}} D_{\text{collected}} \tag{2}$$

Where $f_{\text{preprocess}}$ is the function applied to preprocess the raw data, including filtering, normalization, and formatting.

3.2.3. Data Transmission

$$Dtransmitted = D_{\text{preprocessed}} + E_{\text{transmission}} \tag{3}$$

Where $E_{\text{transmission}}$ accounts for any transmission errors or losses that might occur during data transfer.

3.2.4. Data Reception

$$D_{\text{received}} = Dtransmitted - Ereception \tag{4}$$

3.3. IoT Dashboard Visualization

Once this data is uploaded to the cloud, it appears on an IoT dashboard: the patient's various health parameters are under real-time monitoring to make timely decisions about emergent abnormalities. The dash layout is user-friendly and understandable, with information represented clearly in graph and chart forms. The IoT dashboard receives the transmitted data and updates visualizations with the new data. This, again, ensures an updated system in real-time, so it is important to monitor critical health parameters and communicate any warning signs to the healthcare service provider. For example, if the blood pressure readings show a hypertensive crisis, it sends an alert to indicate immediate medical attention.

The IoT dashboard ensures monitoring of the person's health metrics on an instantaneous basis, thus enabling the continuous updating of critical analysis features that trend, detect anomalies, and alert, underlining significant patterns and thresholds that might help in the early diagnosis of events for the doctor's scrutiny. The visualization process can be mathematically described as follows:

3.3.1. Data Visualization

$$V(t) = g_{\text{visualize}}(D_{\text{soe}}, t) \tag{5}$$

Where $(g_{\text{visualize}})$ is the function that generates visualizations based on the stored data at time t.

3.3.2. Real-Time Update

$$\Delta V(t) = g_{\text{update}}(D_{\text{stored}}, t, \Delta t) \tag{6}$$

Where $\Delta V(t)$ denotes the change in visualization at t, the update function g_{update}, and Δt is the time interval between updates. This ensures continued monitoring and real-time tracking of essential health parameters for early probable health issue detection. Through an IoT dashboard, a patient's health is represented fully based on the patient’s body data in real-time, empowering the healthcare provider with an informed decision and timely intervention. It also uses cloud storage, providing access to data anywhere, enabling remote monitoring and associated telemedicine applications. There are specific security mechanisms in the data transmission of sensitive health data. This is done through data encryption, ensuring that the transmitted information is safe against eavesdropping and sniffing by other people. The process is mathematically represented as

$$Dencrypted = E_{\text{encrypt}} D_{\text{transmitted},K_{\text{public}}} \tag{7}$$

Where (E_{encrypt}) is the encryption function and (K_{public}) is the public key used for encryption.

Decryption at the cloud server end can be represented as:

$$Ddecrypted = E_{\text{decrypt}} D_{\text{encrypted},K_{\text{private}}} \tag{8}$$

Where (E_{decrypt}) is the decryption function and (K_{private}) is the private key used for decryption.

Furthermore, encrypted communication protocols like HTTPS ensure data privacy and confidentiality when transmitting data. So, the general statement of secure transmission can be made as follows:

$$D_{\text{secure}} = \text{HTTPS} D_{\text{encoded}} \tag{9}$$

3.4. Real-Time Data Processing and Alerts

The process continuously analyses the incoming data streams to identify patterns that may represent future health problems. For example, a sudden rise in heart rate or fall in blood oxygen saturation levels may cause alerts. The anomaly detection mechanism can be mathematically exhibited as follows:

$$A(t) = \delta(D(t) - D_{\text{baseline}}) \tag{10}$$

Where the anomaly score $(A(t))$ score at time (t), $(D(t))$ is the current data, and D_{baseline} is the baseline data.

Combined with more advanced data analytics, the health-monitoring system based on the IoT would be much better at predicting and handling health conditions. The system can use deep learning algorithms to analyze past data sets to identify trends and accurate predictions in future health states. Such proactive healthcare involves early intervention and better management of chronic conditions.

3.5. Data Fusion from Multiple Sources

Hence, data fusion techniques are crucial as aggregating data from diverse sources like mobile cars, wearable devices, and stationary sensors adds confidence; however, other acquisition devices are included in the health monitoring system for normal and abnormal patients. Furthermore, data fusion strives to improve health monitoring systems' sturdiness and coverage by merging diverse data streams into a uniform data set.

3.5.1. Data Synchronization

$$D_{\text{synchronized}} = f_{\text{synchronize}}(D_{\text{sensor}}, D_{\text{wearable}}) \tag{11}$$

Where D_{sensor} is the data from stationary sensors, D_{wearable} is the data from wearable devices and $f_{\text{synchronize}}$ is the function that synchronizes the data streams.

3.5.2. Data Aggregation

$$D_{\text{aggregated}} = f_{\text{aggregate}}\left(D_{\text{ssynchronized}}\right) \quad (12)$$

Where $D_{\text{aggregated}}$ represents the aggregated data, offering a holistic perspective on a patient's health.

3.6. LoRa-based Data Transmission

In areas with poor internet, which are thus remote, constant health monitoring cannot depend on the internet for communication. Because of its long-range communications, LoRa technology is well-suited to accomplish this objective. The subsequent portion presents an outline of the implementation of using the Heltec LoRa V2 modules to transmit health sensor data without mobile internet.

3.6.1. Adaptive LoRa Transmission

The proposed system uses an adaptive transmission mechanism termed Adaptive LoRa Transmission. This system transmits sensor data only at the time of abnormal readings, thus saving power consumption and better utilization of bandwidth. It transmits sensor data like SpO2, HR, GSR, BP Sys, BP Dia, and Body Temperature. The Heltec LoRa V2 modules are versatile and powerful tools targeted for long-distance communication. This board integrates an ESP32 microcontroller capable of providing some processing power for handling sensor data and management of communication protocols. This module packs an ISM band LoRa transceiver, mainly in the ISM bands of 868 MHz, giving it many potential applications.

3.6.2. Data Collection and Transmission Process

The data collected is processed in real-time on Raspberry Pi 5, where any anomaly gets detected. If there are abnormal readings, adaptive LoRa transmission dispatches the data using Heltec LoRa V2 modules.

3.6.3. Data Preprocessing

The sensor data is pre-processed to ensure optimal transmission over the LoRa network. This involves filtering the data, normalizing it, and then packetizing it. Packetization ensures that the data segments are appropriately sized for efficient and reliable transmission over the LoRa network.

3.6.4. Packetization

The process of packetization can be modelled as

$$P_i = \text{Packetize}(D_{\text{abnormal}}, i) \quad (13)$$

Where P_i represents the i-th data packet and D_{abnormal} is the abnormal sensor data.

3.6.5. Distance and Range Coverage

The reach, or rather the range, of the Heltec LoRa V2 module is affected by many factors, including environment, antenna design, and transmission power. In an open area, the reach of this module could be 8 km. This might be reduced to some extent when the area is urban and obstruction occurs due to the effects of attenuation on the signal.

The maximum range $R_{\max}$ can be estimated using the following equation:

$$R_{\max} = \frac{c}{4\pi f\sqrt{L}} \quad (14)$$

Where, f is the frequency of operation and L is the system loss factor. With these parameters, the estimated maximum range is:

$$R_{\max} = \frac{3\times10^{8}}{4\pi\times867\times10^{6}\sqrt{L}} \quad (15)$$

The data rate R_b for LoRa communication is determined by the bandwidth B, spreading factor SF, and coding rate CR:

$$R_b = \frac{B\cdot \text{SF}}{2^{\text{SF}}\cdot \text{CR}} \quad (16)$$

The calculation of the data rate for the range of A bandwidth of 125 kHz, along with specific spreading factors and coding rates, gives:

$$R_b = \frac{125\times10^{3}\cdot 12}{2^{12}\cdot\frac{4}{5}} \approx 292\text{bps} \quad (17)$$

Power consumption is one of the most critical considerations in the LoRa modules since it is battery-operated. A baseline for power for the Heltec LoRa V2 modules is with a transmission power of up to +20 dBm and -137 dBm receiver sensitivity. The link budget L_b can be calculated as:

$$L_b = P_t - P_r \quad (18)$$

Where P_t is the transmit power and P_r is the receiver sensitivity.

3.6.6. Data Reception and Processing

Upon reception of the transmitted data, the data is packet processed so that recovery of the initial sensor information is done. The reconstruction further determines the packet integrity and well-ordered packet sequence. This is to ensure that the display on the IoT dashboard is a true reflection of the patient's current health status.

$$D_{\text{received}} = Dtransmitted - Ereception \quad (19)$$

$$Dassembled = \sum_{i=1}^{n} P_i \quad (20)$$

Where Ereception represents any errors or losses during data reception, and Disassembled is the reassembled data from the received packets. The cloud server corrects errors by applying error correction algorithms to ensure the accuracy of data received. This detects and corrects errors that could have occurred during this transmission, improving the system's reliability.

3.6.7. Real-Time Visualization and Alerts

One of the essential features of the health monitoring device is the real-time visualization of sensor data on the IoT dashboard. This independent health metrics view can be attained by the caretaker, with the help of a dashboard, and thus allows continuous monitoring of the patient and reacting to their anomalies in time.

The real-time update mechanism can be mathematically described as follows:

$$\Delta V(t) = g_{\text{update}} D_{\text{asmld},t,\Delta t} \tag{21}$$

Where $(\Delta V(t))$ represents the change in visualization at time t, g_{update} is the update function, (Δt) is the time interval between updates. Therefore, integrating adaptive LoRa Transmission using Heltec LoRa V2 modules is a significant step toward reliable continuous health monitoring, especially in remote areas. It merges the long-range and low-power features of LoRa with real-time data processing and anomaly detection to attain efficient and effective healthcare management. This makes the system very powerful in handling modern healthcare challenges through strategic data preprocessing, packetization, and robust transmission protocols that boost reliability and accuracy.

3.7. Bidirectional LSTM with K-Fold Cross-Validation (BiLSTM-KFCV) for Anomaly Detection

The dataset has been synthesized for deep learning processing; it contains data for 50 patients and 500 health sensor readings for each patient. The detection of anomalies in physiologic data is very vital in health monitoring. The work in this research is carried out by employing a Bidirectional LSTM (BiLSTM) architecture evaluated through a k-fold cross-validation approach, which can capture complex temporal relationships innate in time-series data such as pulse rate, blood pressure (BP), or other vital health signs. The bidirectional LSTM layers designed to process data points would consider the past and future inputs occurring at a specific time step. Bidirectional processing is essential for physiological anomalies that may be well indicated by patterns arising before or after a particular point.

3.7.1. Model Architecture

This model consists of 2 BiLSTM layers with fully connected layers toward the conclusion, ultimately producing a binary classification result influenced by anomaly detection methods. The initial BiLSTM layer contains 64 units and passes the sequence forward to the second BiLSTM layer, which has 32 units, allowing the features to be compacted into a more refined representation. Such a hierarchical structure allows the model first to try to make a very general analysis of the input features and then slowly refocus its attention onto the most salient features for anomaly detection. The dense layers that follow, including one with 50 neurons utilizing the 'ReLU' activation function, help interpret these deep features for the classification task.

3.7.2. Data Preprocessing and Feature Engineering

Most Deep Learning model performance largely relies on how well the input data is organized and processed. The rich dataset for this research consists of different health parameters measured using sensors. Each one of these parameters carries potentially critical information for anomaly detection. Before training, several preprocessing steps were conducted on that data to ingest within the model effectively.

The features have been standardized (whose mean is 0 and variance is set to 1) with the sci-kit-learn library StandardScaler next, and all input features were on a standard scale. Such standardization in deep learning models ensures that all the input features are treated equally and no single feature with a more significant scale takes over the learning dynamics. Following standardization, the input data has been reshaped to be fed according to the expected input format of the LSTM. This involves formatting the dataset into a three-dimensional array, where each instance would be of shape (samples, time steps, features), enabling the LSTM to read the data in that manner as sequences, which is essential to consider for capturing dependencies across time.

$$X_{\text{sae}} = \frac{X-\mu}{\sigma} \tag{22}$$

$$X_{\text{rsae}} = \text{reshape}(X_{\text{sae}}, [\text{samples}, 1, \text{features}]) \tag{23}$$

3.7.3. K-Fold Cross-Validation Approach

To confirm the model's resilience and generalization capabilities, a technique known as K-Fold Cross-Validation is employed with a five-fold setup to balance computational efficiency and validation thoroughness. Here, this technique divides the dataset into five separate groups and, at each step, trains and tests the model iteratively five times. This supports the medical domain since the developed model should perform well for different sub-datasets to be deployed in various real-world scenarios. The latter allows for in-depth logging so that hyperparameter tuning of the model is enabled, and the process is stopped relatively early if degradation of the generalization performance is detected.

This detailed tracking is vital for tuning the model's hyperparameters and for early stopping if the validation performance begins to degrade, a common indication of overfitting.

$$\text{K-Fold Accuracy} = \frac{1}{k}\sum_{i=1}^{k} \text{Accuracy}_i \tag{24}$$

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \tag{25}$$

BiLSTM, through its rigorous training and validation framework, learns features that could help it detect complex anomalies in physiological data—thus asserting the reliance and effectiveness of BiLSTM through a variety of patient data with real-world variability.

Table 2. Pseudo code of Bidirectional LSTM algorithm for health anomaly detection

Algorithm	
Step	**Description**
1	CSV file 'health_monitoring_data.csv' with health data.
2	Target column 'Anomaly'.
3	Model architecture parameters: Two Bidirectional LSTM layers equipped with 64 and 32 units, respectively, each succeeded by a dropout layer set to a 0.2 rate and a single Dense layer containing 50 units.
4	Output layer parameters: 'sigmoid' activation.
5	Training parameters: 50 epochs, 32 batch size.
6	Random state for split: 42.
7	Optimizer: 'adam'.
8	Loss function: 'binary_crossentropy'.
9	Metrics: 'accuracy'.
10	Procedure TRAIN_BIDIRECTIONAL_LSTM_ANOMALY_DETECTION
11	Load Data: Import CSV data into DataFrame.
12	Preprocess Data: Drop 'Anomaly' column, extract target, normalize and reshape features.
13	K-fold Cross Validation Setup: Initialize with 5 splits, shuffle, random state.
14	Build Model: Configure Sequential model with Bidirectional LSTM and Dense layers.
15	Compile Model: Set optimizer, loss function, and metrics.
16	Train Model: Apply the model to the training dataset across each fold, validate.
17	Evaluate Model: Calculate and plot accuracy for each fold.
18	End procedure

Cross-validated performance metrics are provided, and based on these, the reader can find a strong case for how well the model does in its sensitivity or specificity in catching the anomalies that would be of crucial importance for its use in clinical applications.

$$\mathrm{BiLSTM}_{\mathrm{output}}^{(t)} = \mathrm{BiLSTM}\left(X_{\mathrm{rsae}}^{(t)}\right) \tag{26}$$

The training of the BiLSTM model does not only include fitting data into the model. Still, it is accomplished by implementing sophisticated techniques that enhance its learning efficacy and operating robustness. Another regularization method is early stopping, which involves halting the training process when overfitting is detected, typically after the validation set performance fails to improve for two consecutive epochs.

$$\text{Early Stopping Criterion} = (\text{Validation Loss}_{n} - \text{Validation Loss}_{n-1}) \tag{27}$$

Additionally, models in deep learning demand optimization, especially such complex models as BiLSTM, where many hyperparameters need to be tuned. In deep learning, key parameters include the number of LSTM units, the configuration of dense layers, the optimizer, the learning rate, and batch size. The learning rate, in particular, is critical in determining the speed at which the model adjusts to the given task. A learning rate set too high can lead to rapid convergence to a suboptimal solution, while a rate that is too low may result in slow training and an increased risk of getting trapped in a local minimum.

$$\theta_{t+1} = \theta_t - \alpha \nabla_\theta J(\theta_t) \tag{28}$$

3.7.4. Model Validation and Performance Evaluation

On the other hand, model validation has been done by looking at the more excellent picture of model performance in the context of K-Fold Cross-Validation. Every fold arises as an independent test, giving an insight into how the model performs with a new subsection of the dataset. This becomes very relevant in health monitoring, where significant variations in data may occur. Moreover, the ROC-AUC metric quantifies how effectively the model distinguishes between different classes at different threshold settings. A high ROC AUC score would imply that this model can determine quite well between actual presences and absences of anomalies, which is highly important for deploying such models into a clinical environment in which early and accurate detection might significantly change the course of outcomes for patients.

Table 2 presents the pseudo code for the Bidirectional LSTM algorithm designed for health anomaly detection. The process starts with loading the health monitoring data from a CSV file named 'health_monitoring_data.csv', where the target column is labeled 'Anomaly'. The model comprises two bidirectional LSTM layers, featuring 64 units in the first and 32 in the second, and is succeeded by a dense layer containing 50 units. A ‘sigmoid’ activation function is applied at the output layer. The model is trained for 50 epochs with a batch size of 32, and a fixed random state of 42 ensures reproducibility. An optimizer should be the ‘adam’ and loss function ‘binary_crossentropy’ with metric evaluation 'accuracy'. The process should start with importing the CSV data into a DataFrame, dropping the Anomaly column, extracting the target, and normalising and reshaping the features. It then sets a cross-validation procedure utilizing five splits, random shuffling, and a specified random state before building, compiling, training, and evaluating the model to detect anomalies in health data.

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(x)\ d\big(\mathrm{FPR}(x)\big) \tag{29}$$

In summary, the model of Bidirectional LSTM with K-Fold Cross-Validation attains near state-of-the-art accuracy in health monitoring, which integrates deep learning to improve both precision and timeliness of anomaly detection. It pushes forward frontiers in predictive health analytics and sets the groundwork for researching real-time dynamic monitoring health systems. It describes, in detail, this model's architecture, data handling, training, and validation methods that entail complex health data analysis.

3.8. XGBoost Model for Anomaly Detection

The second model for anomaly detection, XGBoost, has been chosen because it has emerged as the most powerful tool for many deep learning challenges, especially in structured and tabular data, a significant output from health monitoring sensors. Thus, it is particularly suitable for processing complex and often heterogeneous data from health monitoring systems. XGBoost works by ensembling many decision trees. Each decision tree, in a sequential way, corrects the mistakes of the previous ones, hence making an improved model to predict the target variable.

3.8.1. Data Preparation and Scaling

The data sample is a collection of health monitoring data from IoT-based sensors based on measures like SpO2, heart rate, GSR, Systolic and diastolic blood pressure (BP measurements for systolic and diastolic values), body temperature, and Respiratory Rate (RR). This implies that the anomalies lie within these parameters, possibly indicators of a health issue. Data preparation involves loading the dataset and feature–target separation: features are health parameters, and the target variable is an anomaly indicator. Features are standardized using StandardScaler to enforce uniform scaling of features, which is mainly handy for the XGBoost classifier.

3.8.2. Model Architecture

The architecture of the XGBoost classifier, referred to here as the "Advanced Health Anomaly Detection Model (AHADM)," is designed to handle the complexity and high dimensionality of health monitoring data. The AHADM employs an ensemble learning approach, specifically leveraging the power of gradient leveraging boosting methods to surpass the predictive capabilities of any single decision tree. The main key to the model's efficacy lies in its ability to sequentially build an ensemble of weak learners sequentially, each compensating for the weaknesses of the previous ones. It enhances the model's collective accuracy and resilience, which is very appropriate for detecting health anomalies from IoT sensor data.

The objective parameter is binary, a binary classification task in that the model predicts an anomaly's presence or absence. The n_estimators would be 100, meaning it builds up the model sequentially to 100 trees. The learning rate is equal to 0.05 and would establish how intense the prediction of each tree. Employing a lower learning rate encourages adding more trees to capture underlying patterns more effectively while diminishing the risk of overfitting. XGBoost constructs trees one after another, in sequence, where every new tree that is built corrects the error of the previous tree in the ensemble that consists of all previously built trees. The error at a given iteration is measured by the following binary cross-entropy employed as the loss function.:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \tag{30}$$

Where ($L(\theta)$) represents the loss function, N indicates the entire number of instances in the dataset, (y_i) refers to the actual label, and ($\hat{y}_i$) corresponds to the predicted probability.

3.8.3. Decision Tree Construction

Decision Tree Construction Each tree makes various data splits, forming a branch to a decision node or a leaf. Such splitting is based on impure criteria in most branches, measured mainly by Gini impurity or entropy criteria. XGBoost aims to minimize a regularized version of the loss function, encompassing the main loss metric and a variant of the regularization penalty to control the model complexity. The regularisation term penalises trees that become overly complex to prevent overfitting.

The algorithm often begins with an initial estimate, such as the average outcome value for the target variable in a regression scenario task or the log-odds value in classification problems. At each iteration, an additional tree is built that helps predict the residuals (error) of the current ensemble. Next, the predictions from this new tree are taken, scaled by the learning rate, and added to the current prediction.

$$\widehat{y_i^{(t)}} = \widehat{y_i^{(t-1)}} + \eta f_t(X_i) \tag{31}$$

$Where,\ \hat{y}^i(t)$ is the updated prediction. $\hat{y}^i(t-1)$ represents the model's forecast generated by the preceding iteration, η is the learning rate, and $ft\ (Xi\ )$ is the prediction from the new tree.

3.8.4. Gradient Boosting Process

In Gradient boosting, a new tree is fitted on the loss function's directional derivative (gradient) about the meantime prediction. This is in contrast to trees being grown individually to overcome the mistakes of the previous ensemble. Generally, it is a gradient boosting that moves towards the direction, which significantly reduces the error. The method used to update each iteration can be mathematically expressed as:

$$\widehat{y_i^{(t)}} = \widehat{y_i^{(t-1)}} - \eta \frac{\partial L(\theta)}{\partial \widehat{y_i^{(t-1)}}} \quad (32)$$

Where $\left(\frac{\partial L(\theta)}{\partial \hat{y}^i(t-1)}\right)$ indicates the direction in which the predictions must be adjusted to minimize the error.

3.8.5. Feature Importance and Interpretation

A vital feature of the AHADM is to have insights into which features are most influential in predicting health anomalies. XGBoost implicitly computes a feature importance score during the training process and indicates the measure of predictive power brought into the model by an input feature.

$$\text{Feature Importance} = \left\{f_1\text{: importance}_1, f_2\text{: importance}_2, \dots, f_n\text{: importance}_n\right\} \quad (33)$$

Where (f_i) represents the $(i) - \text{th importance}$, its corresponding importance score. Visualizing these importances helps identify which health parameters are most critical for detecting anomalies, guiding further investigations and potential interventions.

3.8.6. K-Fold Cross-Validation

The dataset is partitioned into k subsets to improve the dependability of the model. The model is trained on k–1 of these subsets and validated against the remaining one. This procedure is repeated k times so that each subset takes turns serving as the validation set, ensuring thorough evaluation. For this analysis, I select k=5, which will provide a broad estimate of the model quality as k approaches m.

$$\text{KF} = \text{KFold}(n\_splits = 5, shuffle = True, random\_state = 42) \quad (34)$$

The cross-validation method ensures that overfitting of the model is avoided while giving generalization of the evaluation metrics on different dataset splits. The cross-validation technique averages the scores so that it has an overall performance measure.

$$\text{CV Accuracy} = \frac{1}{k}\sum_{i=1}^{k} \text{Accuracy}_i \quad (35)$$

CV Accuracy is the cross-validated accuracy, k denotes the total number of folds, and Accuracy_i represents the accuracy value corresponding to the i-th fold.

Table 3 outlines the pseudo code for the XGBoost algorithm in health anomaly detection. The process begins by loading health data from 'health_monitoring_data.csv', targeting the 'Anomaly' column. The XGBoost model utilizes 100 estimators, maintains a learning rate 0.05, and applies a specified maximum depth of 4. It is designed to perform binary classification using a logistic objective. The dataset is partitioned into dividing the data into 80% for training and 20% for testing while fixing the random state at 42, ensuring consistent results.

3.8.7. Advanced Techniques for Model Improvement

To achieve better performance from the model, some of the advanced techniques for optimization employed include hyperparameter tuning, ensemble strategies, and feature engineering. Hyperparameter tuning refers to the primary key to the optimum performance of the model. Some techniques to explore the hyperparameter space are grid search and random search to identify optimal configurations.

Table 3. Pseudo code of XGBoost algorithm for health anomaly detection

Algorithm	
Step	**Description**
1	CSV file 'health_monitoring_data.csv' with health data.
2	Target column 'Anomaly'.
3	Model architecture parameters: XGBoost with 100 estimators, learning rate 0.05, max depth 4.
4	Output layer parameters: Binary classification with logistic objective.
5	Training parameters: Train/test split 80%/20%.
6	Random state for split: 42.
7	Optimizer: Not applicable.
8	Loss function: 'binary crossentropy' (as logistic regression).
9	Metrics: 'accuracy'.
10	Procedure TRAIN_XGBOOST_ANOMALY_DETECTION
11	Load Data: Import CSV data into DataFrame.
12	Preprocess Data: Optional feature scaling.
13	Split Data: Divide data into training and testing sets.
14	Build Model: Initialize XGBoost classifier with specified parameters.
15	Compile Model: Configure for binary classification.
16	Train Model: Fit model on training data.
17	Evaluate Model: Calculate accuracy on test data.
18	End procedure

The target is to find a combination of hyperparameters that gives a minimum loss function over the dataset.

$$\Theta^* = \arg\min_{\Theta} \mathcal{L}\,(D;\Theta) \tag{36}$$

These key parameters include the learning rate, tree depth, total estimators, subsampling ratio, and the column sampling ratio per tree.

3.9. Random Forest Model for Anomaly Detection

In the context of detecting health anomalies using data from IoT-enabled sensors, the dataset utilized consists of multiple health parameters such as SpO2, heart rate, GSR, and both the systolic and diastolic pressure of blood to detect the anomalies in health from the data of sensors enabled by IoT. The main objective includes accurately identifying anomalous readings, possibly indicating health-related issues. The first step in data handling involves loading the dataset and separating features from the target variable, indicating whether there is an anomaly.

3.9.1. Data Preparation and Feature Standardization

There is no need for feature scaling because the Random Forest model is constructed in such a way that it manages different scales of features due to its tree based structures. However, since common preprocessing steps are suitable not only for uniformity but also to support convergence and increase the effectiveness of the rest of the model-evaluation processes, features are normalized through a StandardScaler to achieve standardization:

$$X_{\text{scaled}} = \frac{X-\mu}{\sigma} \tag{37}$$

Here, (X) represents the matrix of input features, (μ) and (σ) symbolize the dataset's mean and standard deviation, respectively. The features are used to rescale data for zero mean and unit variance. Standardization ensures features are rescaled to produce zero mean and unit standard deviation; tree-based models do not specifically need this, but it does make sure the scale of the data does not inappropriately alter the model evaluation.

3.9.2. Random Forest Architecture for Anomaly Detection

This ensemble learning method works by training numerous decision trees and then giving an output class, the mode of classes predicted by individual trees. The model configuration includes settings for the chosen quantity of decision trees (estimators) and the permitted maximum depth trees, which is a hyperparameter crucial to the capability of the model to generalize:

$$n_{\text{estimators}} = 100, \quad \text{max\_depth} = 10 \tag{38}$$

In this case, 100 is the number of trees used with a controlled maximum depth of 10 to get a suitable balance between bias and variance. A more significant number of trees helps, in general, to boost the model's performance, albeit at the expense of additional computation time and, if not combined with some regularization techniques like having limited tree depth and overfitting.

The Random Forest model is trained by fitting those trees to the training data through an 80% partitioning of the dataset into training and 20% testing sets. This ensures that the model is evaluated on previously unseen data to ensure its performance reflects real-world scenarios:

$$L(\text{model}) = -\sum_{i=1}^{N} y_i \log(p_i) + (1-y_i)\log(1-p_i) \tag{39}$$

In this equation, L(model) defines loss function used during the training, where (y_i) signifies the actual labels, and (p_i) indicates the predicted probabilities for the positive class for each instance$(\,i\,)$. Random Forest minimizes this loss by adjusting the decision thresholds in individual trees.

A Random Forest constructs various decision trees from different sub-samples of input features and instances using random selection. This model training method is known as Bootstrap Aggregating or Bagging and tremendously decreases the model's variance without increasing the bias. The following mathematical expression governs each node-splitting operation in the learning process in each tree:

$$IG(D_p, f) = I(D_p) - \frac{N_{\text{left}}}{N_p} I(D_{\text{left}}) - \frac{N_{\text{right}}}{N_p} I(D_{\text{right}}) \tag{40}$$

Here, $(\,\mathrm{IG}\,)$represents the information gained from feature $(\,f\,)$, $(\,I\,)$, denotes impurity (commonly measured by Gini impurity or entropy), $(D_p)$is the dataset of the parent node, and $(D_{\text{left}})$and(D_{right}) are the datasets of the left and right child nodes, respectively. $(N_p), (N_{\text{left}}), \text{and} (N_{\text{right}})$ correspond to the number of instances in the parent, left child, and right child nodes. Entropy is a measure of impurity or uncertainty in a dataset. It is crucial in determining the best features to split the nodes in a decision tree within a Random Forest:

$$H(S) = -\sum_{x\in X} p(x) \log_2 p\,(x) \tag{41}$$

This equation calculates the entropy $(H(S))$of a set$(\,S\,)$, where $(\,X\,)$ represents the classes in$(\,S\,)$, and $(p(x))$ is the probability of class $(\,x\,)$ appearing in set$(\,S\,)$. Information gain is employed to determine the optimal feature for partitioning at each stage in the tree:

$$IG(T, X) = H(T) - \sum_{i=1}^{n} \frac{|T_i|}{|T|} H(T_i) \tag{42}$$

Here, $(\mathrm{IG}(T, X))$ is the information gained from splitting tree $(\,T\,)$ using feature (X), $(H(T))$ is the entropy of $(\,T\,)$ before the split, (T_i) are the subsets of $(\,T\,)$ after the split, $(|T_i|/|T|)$ is the proportion of the number of elements in $(T_i)$to(T).

For each tree in the forest, a recursive choice of best split among all features, by maximum information gain, is considered up to a maximal depth or until no further information gain can be achieved. The ultimate decision made by the ensemble is through a weighted summation of all the individual trees' predictions.

$$y_{\text{pred}} = \frac{1}{n_{\text{trees}}} \sum_{i=1}^{n_{\text{trees}}} T_i(x) \tag{43}$$

Where $(T_i(x))$ represents the prediction of the (i)-th tree, and (x)is the input feature vector.

3.9.3. Advanced Feature Engineering in Random Forest

$$X_{\text{new}} = X_i \times X_j, \quad \forall i, j \in \{1, \dots, n\}, i \neq j \tag{44}$$

Equation (44) represents the interaction between different sensor readings, in which $(X_i) and (X_j)$ are the individual sensor features. By adding these interaction terms, the model can learn from a particular sensor reading and how different readings relate to one another, which, in most cases, is critical to accurate anomaly detection.

Moreover, polynomial features of sensor readings are generated to capture more complex non-linear relationships:

$$X_{\text{poly}} = (X_i)^d, \quad \forall i \in \{1, \dots, n\}, d = 2,3, \tag{45}$$

Here, $((X_i)^d)$denotes the (d)-th power of sensor reading (X_i), allowing the model to fit higher-degree polynomial relationships in the data.

The features in the Random Forest can be computed as a decrease in node impurity multiplied by the probability of reaching that node (or the proportion of samples that go through that node).

$$\text{Imp}(X_m) = \sum_{t \in \text{Trees}} \sum_{n \in t: \text{ splits on } X_m} p(n) \cdot \Delta i(n, X_m) \tag{46}$$

Where $(\text{Imp}(X_m))$ is the importance of features (X_m), (t) enumerates over the trees, (n) enumerates over the nodes in the tree (t) that split on the feature (X_m), $(p(n))$ is the proportion of samples that reach node n, and (n), and$(\Delta i(n, X_m))$ is the change in impurity from splitting on the feature (X_m) at node (n). This should be added to the section discussing feature importance to explain how each feature's contribution to the model is quantified.

3.9.4. Detailed Analysis of Model Training and Node Splitting

In the Random Forest model, $\widehat{y_{\text{pred}}}$ aggregates predictions $T_i(x)$ from N trees, each trained on unique data subsets through bootstrapping, enhancing prediction accuracy and robustness against overfitting. This ensemble method effectively combines multiple weak learners, mitigating overfitting by averaging their predictions and ensuring comprehensive data coverage.

$$y_{\text{pred}} = \frac{1}{N} \sum_{i=1}^{N} T_i(x) \tag{47}$$

3.9.5. Model Development and Analysis

As this robust health monitoring model develops, Further enhancements involve hyperparameter tuning via grid search and random search, advanced metrics AUPRC for improved model performance evaluation, and the continuous incorporation of sensor data as they become available to enhance and validate model predictions. All these steps were necessary to ensure that the Random Forest model is highly accurate and gives trusted, reliable, and actionable information for use in healthcare.

Table 4 illustrates the pseudo code of using a Random Forest algorithm for health anomaly detection. It is initiated by reading health data from 'health_monitoring_data.csv', with its target column named 'Anomaly'. The model is implemented with a Random Forest based on 100 trees and a maximum depth 10. Data will now be split for 80% train and 20% test. A Random state of 42 has been assigned for reproducibility purposes.

3.10. Long Short-Term Memory (LSTM) Model for Health Risk Prediction

Long Short-Term Memory (LSTM) networks, a specialized Recurrent Neural Network (RNN), are particularly effective for sequence-based predictions, making them especially useful for health risk prediction steps. This work applies LSTM models in the health data analysis of a time series nature, with various sensors, to calculate hypertension, hypoxia, cardiac stress, fever conditions, and stress.

With different approaches, LSTMs can easily handle long-term dependencies or patterns within the data, making it feasible to provide accurate forecasting for health trends, a critical phase in proactive health management. Equipped with such advanced capabilities of LSTM to provide early warnings with actionable insights that could greatly benefit patients regarding their outcomes and facilitate timely medical interventions. This implementation highlights how deep learning models can transform health monitoring systems by offering predictive analytics in support of preventive strategies on health.

3.10.1. Preprocessing

An IoT-enabled health monitoring system embeds enough preprocessing steps within the collection process. This incorporates numerous sensors explicitly developed to continuously measure most of a patient's physiological parameters: systolic and diastolic blood pressure (BP), pulse rate, oxygen saturation (SpO2), body temperature, galvanic skin response, and electromyography sensors.

Table 4. Pseudo Code of Random Forest algorithm for health anomaly detection

Algorithm	
Step	**Description**
1	CSV file 'health_monitoring_data.csv' with health data.
2	Target column 'Anomaly'.
3	Model architecture parameters: RandomForest with 100 trees, max depth 10.
4	Output layer parameters: Not applicable.
5	Training parameters: Train/test split 80%/20%.
6	Random state for split: 42.
7	Optimizer: Not applicable.
8	Loss function: Default for RandomForest.
9	Metrics: 'accuracy', 'ROC AUC Score'.
10	Procedure TRAIN_RANDOM_FOREST_ANOMALY_DETECTION
11	Load Data: Import CSV data into DataFrame.
12	Preprocess Data: Optional feature scaling.
13	Split Data: Divide data into training and testing sets.
14	Build Model: Initialize RandomForest classifier with specified parameters.
15	Compile Model: Set metrics.
16	Train Model: Fit model on training data.
17	Evaluate Model: Calculate accuracy, generate classification report, and compute ROC AUC score.
18	End procedure

A crucial step to put through a preprocessing phase after data collection involves cleaning up the data and returning it to a normalized, properly formative, and qualitatively usable form in preparation for model training. Data cleaning is the first thing that needs to be done before preprocessing, and it is the process of removing noise and outliers introduced during the data collection.

The next step in the preprocessing pipeline is normalization, which is crucial because it puts data on a consistent scale. This process directly influences the effectiveness of deep learning models. The extracted features from the dataset are BP_Sys, BP_Dia, HR, SpO2, Temp, GSR, and EMG, for which standard scaling is done using the feature-column standard scalar. This scaling procedure subtracts the mean and normalizes each feature to a variance of one. Normalization is mathematically represented as:

$$X_{\mathrm{sae}} = \frac{X-\mu}{\sigma} \tag{48}$$

Where, X is the feature matrix, μ is the mean of X, and σ is the standard deviation of X, which keeps all features to have mean = 0 and standard deviation = 1, leading to faster learning and convergence of deep learning models. The data is reshaped after normalization. This is required to meet the input expectations of the LSTM. LSTM networks take 3D input tensors, where the dimensions are samples, time steps, and features.

$$X_{\mathrm{sae}} = X_{\mathrm{sae}}.reshape\big((X_{\mathrm{sae}}.shape[0], 1, X_{\mathrm{sae}}.shape[1])\big) \tag{49}$$

This transformation ensures the data is properly formatted so that the LSTM network can efficiently capture and process temporal dependencies. The final step in the preprocessing pipeline involves dividing the dataset into training and test sets. This now becomes important; this way, when the model is tested on new data, it gives an unbiased estimate of its performance. 80% of the data is fed into the train, and the rest is fed for testing. Therefore, data collection and preprocessing play a key role in model training and evaluation. These phases significantly improve performance and enhance the reliability of deep learning models about IoT-empowered health monitoring by ascertaining uniformity and correct scaling. The data preprocessing techniques use the best quality data to be integrated with predictive modeling for health applications, producing sound results and well-timed and accurate predictions of health risks.

Therefore, the multi-sensor data used in the model are very important in developing an all-inclusive health monitoring system. Different information possessed by each category of sensor data about a patient's physiological state assists the model to accommodate comprehensive health indicators. Blood pressure measurements, both systolic and diastolic, are two of the major defining indicators of hypertension and hypotension. Essentially, these measures are critical in establishing cardiovascular health. Monitoring heart rate is important because variations may suggest cardiac stress or even an underlying arrhythmia and detect abnormal heart rhythms. Monitoring body temperature is also a basic health parameter for identifying fever and possible infections, which could indicate more serious underlying health issues. The Galvanic Skin Response (GSR) quantifies the skin's electrical conductivity, which varies according to sweat gland activity. Again, it is related to the state of stress and emotion and hence indicative of psychological conditions. Another diagnostic tool, Electromyography, records muscle response and activities. It is particularly useful in diagnosing neuromuscular disorders and an overview of muscle health in the comprehensive assessment of muscular functions.

3.10.2. Model Architecture and Training

This IoT-enabled system predicts health issues in advance using different deep-learning models based on data from real-time sensors attached to the body. This paper proposes a deep, Long Short-Term Memory Network specifically tailored to work with time series sensor data and infer activities from physiological sensor data. The approach uses an LSTM model that can build on temporal dependencies in physiological data through data collected via various sensors. Key physiological parameters include systolic and diastolic blood pressure, heart rate, oxygen saturation levels, and temperature. All the independent variables or input features (X) will predict future health outcomes. Predicted risk conditions are future hypertension, future hypoxia, future cardiac stress, future fever, and future stress.

The LSTM model is defined with an input layer that takes preprocessed and reshaped sensor data. It has an input layer, after which two stacked LSTM layers are presented, and two dropout layers of 0.2 are subjected to each of the two LSTM layers. The initial LSTM layer consists of 50 units and is configured to return sequences, stacking another LSTM layer. The design can use the model to capture complex patterns over time on sequential data. A second LSTM layer, also with 50 units, has been added to handle the data sequentially. To avoid overfitting, dropout layers of 20% were also placed between the LSTM layers to improve the model's generalisation.

$$H\_0 = Input(shape = (1, X\_train.shape[2]))$$

$$H\_1 = LSTM(50, return\_sequences = True)(H\_0)$$

$$H\_2 = Dropout(0.2)(H\_1)$$

$$H\_3 = LSTM(50)(H\_2)$$

$$H\_4 = Dropout(0.2)(H_3) \tag{50}$$

The final layers of the model are five dense output layers, each corresponding to one of the predicted health conditions. Each output layer employs a sigmoid activation function to generate a probability score corresponding to its respective condition. The output layers are defined as follows:

$$output\_hypertension = Dense(1, activation = 'sigmoid', name = 'output\_hypertension')(H\_4) \tag{51}$$

$$output\_hypoxia = Dense(1, activation = 'sigmoid', name = 'output\_hypoxia')(H\_4) \tag{52}$$

$$output\_cardiac\_stress = Dense(1, activation = 'sigmoid', name = 'output\_cardiac\_stress')(H\_4) \tag{53}$$

$$output\_fever = Dense(1, activation = 'sigmoid', name = 'output\_fever')(H\_4) \tag{54}$$

$$output\_stress = Dense(1, activation = 'sigmoid', name = 'output\_stress')(H\_4) \tag{55}$$

During training, the iterative updates of model weights to reduce the loss function effectively. In this process, the training has been closely monitored to avoid the overfitting problem in the model on the training data, which could be done by including dropout layers. The dropout layers are essential in maintaining a model's generalisation capability, achieved through randomly dropping units during the training procedure. The model architecture of sequential data processing is such that it is utilized to mine the inherent temporal relationships within the physiological readings. The LSTM layers can learn patterns across time, which is considered most important in predicting conditions like hypertension and hypoxia. Including dropout layers decreases the risk of overfitting, so the projections can remain reliable even under new, unseen data.

3.10.3. Forget Gate

$$f_t = \sigma\left(W_f \begin{bmatrix} BP_{Sys_{t-1}}, HR_{t-1}, SpO2_{t-1}, Temp_{t-1}, \\ GSR_{t-1}, EMG_{t-1} \end{bmatrix} + b_f\right) \tag{56}$$

The forget gate decides what part of the information from the previous time step should be discarded: where previously, systolic blood pressure (BP_Sys), heart rate (HR), oxygen saturation (SpO2), body temperature (Temp), galvanic skin response (GSR), and electromyography (EMG) were considered, and the activation of the forget gate, (f_t) using the corresponding weights (W_f) and bias (b_f).

3.10.4. Input Gate

$$i_t = \sigma\left(W_i \begin{bmatrix} BP_{Sys_{t-1}}, HR_{t-1}, SpO2_{t-1}, Temp_{t-1}, GSR_{t-1}, \\ EMG_{t-1} \end{bmatrix} + b_i\right) \tag{57}$$

The input gate determines which information from the current input should be written to the cell state. It uses the previous time step values of the physiological parameters to compute the input gate activation (i_t) with weights (W_i) and bias (b_i).

$$\widetilde{C_t} = \tanh\left(W_C\begin{bmatrix}BP_{Sys_{t-1}}, HR_{t-1}, SpO2_{t-1}, Temp_{t-1}, \\ GSR_{t-1}, EMG_{t-1}\end{bmatrix} + b_C\right) \quad (58)$$

The candidate memory cell ($\widetilde{C_t}$)generates new candidate values that could be added to the cell state, considering the previous values of the physiological parameters. This operation uses the hyperbolic tangent function ((tanh)) to squash the values between -1 and 1, with weights (W_C) and bias (b_C).

3.10.5. New Memory Cell

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \quad (59)$$

The new cell state (C_t) is updated based on the forget gate (f_t), the previous cell state (C_{t-1}), the input gate (i_t), and the candidate memory cell ($\widetilde{C_t}$). The old values of the physiological parameters calculate the activation with weights at the output gate.

3.10.6. Output Gate

$$o_t = \sigma\left(W_o\left[BP_{Sys_{t-1}}, HR_{t-1}, SpO2_{t-1}, Temp_{t-1}, GSR_{t-1}, EMG_{t-1}\right] + b_o\right) \quad (60)$$

The output gate (o_t) decides what part of the cell state should be output. It uses the previous values of the physiological parameters to compute the output gate activation with weights (W_o) and bias (b_o).

3.10.7. Hidden State

$$h_t = o_t * \tanh(C_t) \quad (61)$$

The hidden state (h_t) is the output of the calculation using the output gate (o_t) and the updated cell state (C_t). This hidden state is also passed to the next time step and used for prediction.

3.10.8. Dense Layer Output

$$y_t = \sigma\left(W_y h_t + b_y\right) \quad (62)$$

The final output from the dense layer is obtained using the hidden state , weights and bias. This output represents the predicted probability of the respective health condition, such as hypertension, hypoxia, cardiac stress, fever, or stress.The final output (y_t) from the dense layer is calculated using the secret state (h_t), weights (W_y), and bias (b_y). This is the result or prediction probability of the corresponding health condition like high blood pressure, low oxygen level, high working of heart, fever, or stress.Table 5 outlines the pseudo code for a Multi-Output LSTM model designed to predict health conditions. The process begins by loading enhanced health data from the dataset. Input features include 'BP_Sys', 'BP_Dia', 'HR', 'SpO2', 'Temp', 'GSR', and 'EMG', with target conditions such as 'Future_Hypertension' and 'Future_Hypoxia'. Features are normalized using StandardScaler and reshaped for LSTM compatibility. The dataset is split into 80% for training and 20% for testing, ensuring reproducibility with a random state of 42. The LSTM model architecture is defined with multiple output layers. The model employs the 'Adam' optimizer, uses binary cross-entropy as the loss function, and evaluates performance using the accuracy metric.The procedure includes loading and preprocessing the data, building and compiling the model, fitting it for 50 epochs with 32 batch sizes and testing its performance on test data.

Table 5. Pseudo Code of Multi-Output LSTM Model for Predicting Health Conditions

Algorithm	
Step	**Description**
1	CSV file '/content/health_monitoring_data_enhanced.csv' with enhanced health data.
2	Input features 'BP_Sys', 'BP_Dia', 'HR', 'SpO2', 'Temp', 'GSR', 'EMG'.
3	Target conditions 'Future_Hypertension', 'Future_Hypoxia', 'Future_Cardiac_Stress', 'Future_Fever', 'Future_Stress'.
4	Normalize features using StandardScaler and reshape for LSTM compatibility.
5	split them into 80% for training and 20% for testing with random state 42.
6	Define LSTM model architecture with multiple output layers.
7	Optimizer: 'adam'.
8	Loss function: 'binary_crossentropy'.
9	Metrics: 'accuracy'.
10	Procedure TRAIN_MULTI_OUTPUT_LSTM
11	Load Data: Import CSV data into DataFrame.
12	Preprocess Data: Extract features and targets, normalize and reshape.
13	Build Model: Configure input, LSTM, dropout, and output layers.
14	Compile Model: Set optimizer, loss function, and metrics.
15	Train Model: Fit model on training data for 50 epochs with a batch size of 32.
16	Evaluate Model: Assess model performance using test data.
17	End procedure

3.11. Gated Recurrent Unit (GRU) Model for Health Risk Prediction

Data preprocessing is crucial in preparing the dataset to train the GRU model, ensuring that the sensor data is learning effectively. Physiological metrics include systolic and diastolic blood pressure, heart rate (HR), oxygen saturation (SpO_2), body temperature, Galvanic Skin Response (GSR), and electromyography (EMG) obtained from the sensors is explained in this section. The raw sensor readings would be transformed into a format allowing input to the GRU model to make relatively better predictions of health risk conditions.

One approach to preprocessing is to clean the raw sensor data: handle missing values, detect and remove outliers, and smooth data to reduce noise. This is a common problem in collecting real-time sensor data due to malfunctions or transmission errors, which cause missing values. Techniques for handling missing values include mean and median imputation so that the dataset is not left empty but complete and consistent. Outliers disrupt the learning process of a model. These are detected as outliers by statistical methods and either removed or replaced with some more representative value.

Normalization follows data cleaning, during which the data has had noise removed from it. Normalization scales the data to lie in a standard range, commonly between 0 and 1. This procedure is crucial for the efficiency of deep learning models, often utilized with a Normalizer or StandardScaler from the popular Python library scikit-learn. The normalization equation can be expressed as:

$$X_{\mathbf{normalized}} = \frac{X-\mu}{\sigma} \tag{63}$$

Where X denotes the feature matrix, μ indicates the mean of X, and σ represents the standard deviation of X.

Next process is reshaping data into the required input shape concerning the GRU network. The input tensors used in a GRU network are 3D, as shown by several samples, time steps, and features. In this regard, the reshaping process ensures that the data structure conforms to proper arrangements that allow a GRU to deal with temporal dependencies effectively.

$$X_{\text{rsae}} = X_{\text{nraie}}.reshape\big((X_{\text{nraie}}.shape[0], 1, X_{\text{nraie}}.shape[1])\big) \tag{64}$$

These preprocessing steps turn the raw data into a structured, normalized format for better learning capabilities from the data in the GRU model and improved predictions. Clean, normalized, and reshaped data ensure the model captures the physiological parameters' underlying temporal dependencies and further results in reliable health condition prediction.

3.11.1. GRU Model Architecture

The architecture of the Gated Recurrent Unit (GRU) model is well-crafted to consider sequential data, which is an excellent choice for deriving inferences on any kind of physiological signal extracted from different types of sensors. The proposed architecture can accurately capture the time dependencies in the data and yield good predictions concerning health conditions like hypertension, hypoxia, cardiac stress, fever, and stress. Reduced complexity in the gating mechanism, without compromise in handling the long-term dependencies, renders it very apt for real-time health monitoring applications. The model begins using an input layer accepting normalized and reshaped sensor data. Data is input into the input layer in three dimensions: sample, time steps, and features. The structure of the input layer needs to be defined to fit preprocessed data so that it will be 'cleanly' fed to the GRU layer below. This is important for the first layer, which is the entry point for the physiological data from the sensors.

$$H_0 = \text{Input}\big(shape = (1, X_{\text{train}}.shape[2])\big) \tag{65}$$

3.11.2. The Role of the GRU Layer

The architecture holds a GRU layer of 50 units as the core. Unlike LSTM, in GRU, the forget and input gates are combined into a single update gate. This makes it much simpler, though effective. It makes the training process quicker and the model faster. The GRU layer processes the input sequences, capturing long-range dependencies and learning the temporal patterns in physiological data.

$$H_1 = \text{GRU}(50,\text{return\_sequences=False})(H_0) \tag{66}$$

The entire sequence is run through the GRU layer to output the final hidden state. This means further encapsulating learned features from an input sequence passed into it and then forwarding it to a dense layer. The dense layer, which has 50 units, is then activated by the ReLU function, a non-linear activation that allows the neural network to be trained on very complex patterns.

$$H_2 = \text{Dense}(50,\text{activation='relu'})(H_1) \tag{67}$$

Five dense output layers for predicting the probability of each health condition. Each output layer corresponds to one health condition being monitored: hypertension, hypoxia, cardiac stress, fever, and stress. These layers use sigmoid activation functions to compute probability scores between 0 and 1, which measures how likely each condition is true. This architecture ensures that the model provides interpretable and actionable predictions on each health metric.

$$"\{outputs\} = ["\{Dense\}(1,"\{activation = 'sigmoid'\},"\{name = f'output\_\{i\}'\})(H\_2) "\{\ for\ \} i "\{\ in\ range\}(5)] \tag{68}$$

The update gate z_t is an important factor that determines how much of the past information should be carried forward to the future.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad (69)$$

3.11.3. Reset Gate Mechanism

The mechanism of reset gate determines how much of the past information will be forgotten. This gate ensures selective remembrance, i.e., parts of the prior hidden state are to be forgotten. This is ensured by enabling the model to emphasise only the most essential features at a given time step.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (70)$$

3.11.4. Candidate Hidden State

The reset gate is then used to compute a candidate hidden state $\widetilde{h_t}$, which regulates the past hidden state before passing through the tanh activation. In turn, this state contains both the present input and past hidden state after adjustments, hence becoming the foundation of a new hidden state.

$$\widetilde{h_t} = \tanh(W_h \cdot [r_t * h_{t-1}, x_t] + b_h) \quad (71)$$

3.11.5. Transition to Dense Layers

After the GRU layer, a dense layer of 50 units and the activation function ReLU were added. The next dense layer has 50 units, which provides non-linearity to the model to avoid the vanishing gradient problem by enhancing learning capability.

The ReLU activation function helps avoid the vanishing gradient problem, enhancing the model's learning capabilities.

$$H_2 = \text{Dense}(50,\text{activation='relu'})(H_1) \quad (72)$$

3.11.6. Output Layers for Health Condition Prediction

The model's last stage includes five dense output layers associated with a specific health condition: hypertension, hypoxia, cardiac stress, fever, and stress. Sigmoid activation functions at these layers generate probability scores between 0 and 1. Inference from the probabilistic output regarding the likelihood of each health condition could interpret timely medical interventions.

$$"\{outputs\} = ["\{Dense\}(1,"\{activation = 'sigmoid'\},"\{name = f'output\_\{i\}'\})(H\_2)\ "\{\ for\ \}\ i\ "\{\ in\ range\}(5)] \quad (73)$$

3.11.7. Final Hidden State Calculation

The output hidden state $(\mathrm{h_t})$ is formed by the previous hidden state $(\mathrm{h_{t-1}})$ and the candidate's hidden state$(\widetilde{\mathrm{h_t}})$ being combined with the update gate $(\mathrm{z_t})$.

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t} \quad (74)$$

3.11.8. Dense Layer Activation

The dense layer activation $(\mathrm{d_t})$ applies the ReLU activation function to the current hidden state $(\mathrm{h_t})$, enabling the model to capture complex data relationships.

$$d_t = \text{ReLU}(W_d \cdot h_t + b_d) \quad (75)$$

3.11.9. Health Condition Outputs

Each output layer computes the probability of a specific health condition using the dense layer activation $(\mathrm{d_t})$ and a sigmoid function, ensuring the outputs are in the form of probabilities.

Hypertension

$$y_{t,\text{hypertension}} = \sigma(W_y \cdot d_t + b_y) \quad (76)$$

Hypoxia

$$y_{t,\text{hypoxia}} = \sigma(W_y \cdot d_t + b_y) \quad (77)$$

Cardiac Stress

$$y_{t,\text{cardiac\_stress}} = \sigma(W_y \cdot d_t + b_y) \quad (78)$$

Fever

$$y_{t,\text{fever}} = \sigma(W_y \cdot d_t + b_y) \quad (79)$$

Stress

$$y_{t,\text{stress}} = \sigma(W_y \cdot d_t + b_y) \quad (80)$$

The architecture design empowers the GRU model in capturing temporal dependencies and nonlinear relations within the physiological data. This is quite critical in real-time health monitoring, where one is called upon to make a continuous and accurate prediction for timely medical interventions. Besides, GRU units enable the efficient handling of long sequences, hence making the model suitable for monitoring chronic conditions that evolve over time.

3.11.10. Model Evaluation

The GRU-based health monitoring system has been evaluated for reliability and accuracy in predicting many health conditions. This process involved evaluating predictive accuracy and loss metrics analysis and interpreting results into real-world applications. The main concern of this section is the methods and metrics used to evaluate the GRU model, hence detailing the information on its effectuality. These include accuracy, precision, recall, F1 score, and support. These metrics provide multidimensional information about the performance of the GRU model in such a way that overall metrics indicate areas of strength and weakness. This suite of metrics allows for a broad analysis of the model's efficacy under various health scenarios. Table 6 outlines the pseudocode for a GRU (Gated Recurrent Unit) model employed to predict future health conditions based on physiological parameters.

Table 6. Pseudo code of GRU model for predicting future health conditions

Algorithm	
Step	**Description**
1	CSV file '/content/health_monitoring_data_enhanced.csv' with enhanced health data.
2	Input features 'BP_Sys', 'BP_Dia', 'HR', 'SpO2', 'Temp', 'GSR', 'EMG'.
3	Target conditions 'Future_Hypertension', 'Future_Hypoxia', 'Future_Cardiac_Stress', 'Future_Fever', 'Future_Stress'.
4	Normalize features using StandardScaler and reshape for GRU compatibility.
5	split the dataset into 80% for training and 20% for testing with random state 42.
6	Define GRU model architecture with a single output for each condition.
7	Optimizer: 'adam'.
8	Loss function: 'binary_crossentropy'.
9	Metrics: 'accuracy'.
10	Procedure TRAIN_GRU_MODEL
11	Load Data: Import CSV data into DataFrame.
12	Preprocess Data: Extract features and targets, normalize and reshape.
13	Build Model: Configure input, GRU, and output layers.
14	Compile Model: Set optimizer, loss function, and metrics.
15	Train Model: Fit model on training data for 50 epochs with batch size of 32.
16	Evaluate Model: Assess model performance using test data and compute classification reports.
17	End procedure

The predictive process begins by loading the enhanced health dataset, containing a comprehensive set of features and target conditions. The key input features include 'BP_Sys' (systolic blood pressure), 'BP_Dia' (diastolic blood pressure), 'HR' (heart rate), 'SpO2' (blood oxygen saturation), 'Temp' (body temperature), 'GSR' (galvanic skin response), and 'EMG' (electromyography readings). The target variables, 'Future_Hypertension' and 'Future_Hypoxia', represent potential future health risks, enabling early interventions and proactive health management.

Before model training, the data undergoes preprocessing to ensure compatibility with the GRU architecture. The preprocessing steps include normalizing the input features using the StandardScaler, which scales the features with a mean of 0 and a standard deviation of 1. This normalization step is critical for ensuring optimal performance of the GRU model, as it helps stabilize the training process. The normalized data is then reshaped into a 3D tensor format required by GRU layers, where the dimensions correspond to the sample size, time steps, and number of features.

The dataset is split into training and testing subsets, with 80% allocated for training and 20% for testing. The split ensures the model has sufficient data to learn from while maintaining a separate evaluation set to measure its generalization capabilities. A random seed (random state of 42) is set to ensure reproducibility of results.The GRU model architecture is designed for binary classification tasks, with a single output neuron for each target condition. The model leverages GRU layers to capture temporal dependencies in the health data, followed by a dense layer with a sigmoid activation function to output probabilities for the binary classification. The 'adam' optimizer is used for its adaptive learning rate and efficient performance on large datasets. The model minimizes the 'binary_crossentropy' loss function, which is suitable for binary classification tasks, and uses 'accuracy' as the evaluation metric. The model training process involves fitting the GRU model on the training data for 50 epochs using mini-batches with a batch size of 32. Mini-batching helps in efficient memory usage and faster computation, especially when dealing with large datasets. During training, the model iteratively updates its weights to minimize the loss function, improving its ability to predict future health conditions. The GRU (Gated Recurrent Unit) model serves a similar purpose to the LSTM model but is often more efficient due to its simpler architecture. This model predicts the same set of future health conditions by processing input features through a GRU layer, which helps reduce complexity and improve training efficiency compared to LSTMs.

3.11.11. Ethical and Regulatory Compliance

Ensuring patient data privacy and regulatory adherence is critical to healthcare systems. The proposed IoT-enabled health monitoring system incorporates robust privacy-preserving measures to protect sensitive health information. First, all sensor data is encrypted using the AES-256 encryption standard during transmission and storage, preventing unauthorized access. This encryption ensures that health data remains secure across all phases of operation. Second, patient identifiers are anonymized before data transmission to comply with healthcare regulations, ensuring that personal information is not exposed. This approach aligns with global standards such as the Health Insurance Portability and Accountability Act (HIPAA) in the United States and the

General Data Protection Regulation (GDPR) in the European Union. Secure communication protocols such as HTTPS and Transport Layer Security (TLS) are employed to enhance data security further.

These protocols maintain data integrity and confidentiality by establishing secure channels for data transmission. Additionally, the system strictly adheres to healthcare regulations and standards, including HIPAA for ensuring health data privacy, GDPR for protecting personal data, and ISO/IEC 27001, which outlines best practices for information security management.

4. Results and Discussions

This section provides detailed information on sensor-based health monitoring systems embedded with sensors for real-time data visualization and efficient management of health-related issues. The results show that adaptive LoRa communication assures data transmission even upon the Internet's loss by optimizing energy consumption and operational cost. The results present the high accuracy and precision of deep learning models comprising BiLSTM, XGBoost, and GRU in anomaly detection and health risk prediction. These findings show that a system can improve remote health monitoring by providing essential insights and timely medical interventions.

4.1. Integration of IoT Sensor Data with Real-Time Health Monitoring Systems

This transformation framework uses the prevention, diagnosis, and management of an illness as health services and puts them into practical use based on technologies from the Internet of Things (IoT). This research focuses on the innovative IoT-based health monitoring system, which integrates various sensors that can collect real-time data concerning systolic and diastolic blood pressure, heart rate, oxygen saturation, galvanic skin response, electromyography, temperature of the body, ECG and particulate matter levels. These parameters are displayed on the dashboard in real-time, rendering insights into the real-time status for necessary early interventions to manage health effectively.

4.1.1. Real-Time Data Visualization

The IoT dashboard is very vital for the effective representation of health data. It detects readings from all sensors and draws a correct analysis of patient health in detail, which may raise an alarm or quick concern during detecting every abnormality. Its design ensures that all data from each sensor refreshes in real time so that its interpretation and decision-making can be intuitive. Customized IoT dashboard visualization is used for this purpose. It offers powerful tools for visualizing sensor data through customizable and interactive charts, graphs, and alerts by leveraging a proprietary IoT dashboard that seamlessly integrates various sensor inputs, presenting them in a coherent, user-friendly format.

4.1.2. Adaptive LoRa Transmission in Health Monitoring IoT Systems

Table 7. LoRA Communication

Parameter	Specification
Data Speed	5 kbps
Bandwidth	125 kHz
Range	6 Km (Semi Urban Area)
Transmission Power	Up to 20 dBm
Network Availability	99%
Packet Loss	< 1%
Latency	300 ms
Error Rate	0.2%

Adaptive LoRa communication in IoT health monitoring has a great and relevant solution built on the needs of dynamic requirements for the transmission of healthcare data. This research demonstrates the use of adaptive LoRa modules with HeltecV2 modules to transmit critical health information in urban areas effectively. The system runs only on abnormal sensor readings, guaranteeing efficient and timely medical interventions.

4.1.3. System Configuration and Performance Metrics

A self-adaptive LoRa system has been implemented, where its parameters, like the data rate and transmission power, are self-adjusting based on variations in network conditions and the importance of data being transmitted by these sensors. Selective transmission of health data in the adaptive LoRa system, due to turning it on and off while detecting anomalies, increases the entire system's efficiency, avoiding unnecessary data traffic. Healthcare providers only need alerts when immediate medical response attention is required. The adaptive LoRa setup offers 30% better energy efficiency than non-adaptive systems, which is a feasible solution for continuous health monitoring with reduced operational costs and prolonged lifetime of the battery-operated health sensors.

4.2. LSTM on Anomaly Detection

A Bidirectional LSTM model for detecting health data anomalies from Internet-of-Things sensors is proposed. In this deep learning approach, excellent performance can be shown in the metrics in which methods capture necessary temporal dependencies critical for identifying physiological signal anomalies. Figure 2 represents the confusion matrix, the key summary output of model performance that reveals how good it is at correctly classifying instances. With a total of 6,573 true positives and 5,916 true negatives, the model displays high accuracy in identifying normal and anomalous states. The other areas where it stands out are the low numbers for false negatives at eight and false positives at three. In such applications, medical ones being prime examples, the cost of the false negatives is sometimes the price one pays with one's life. A high value of true-positive rate ultimately reveals the sensitivity of the model, which is an essential aspect of medical diagnostics, in such a manner that minimal anomalies are being missed.

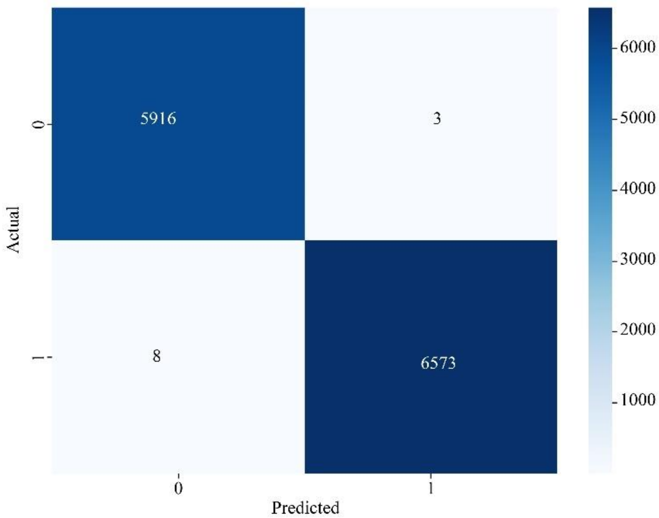

Fig. 2 Confusion matrix

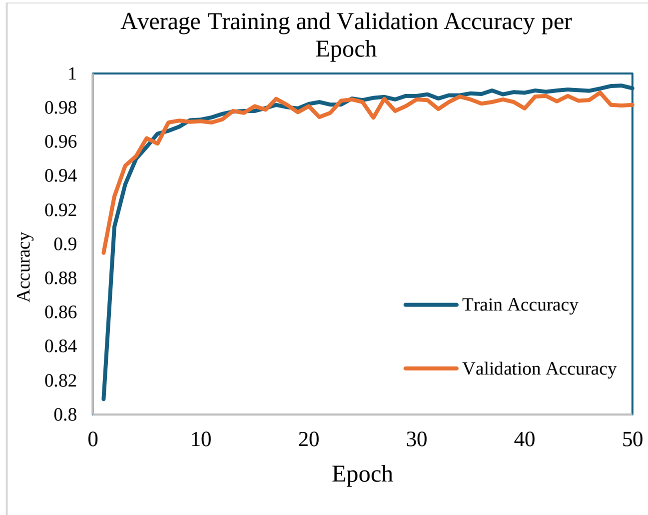

Fig. 4 Average training and validation accuracy

4.2.1. Anomaly Score Distribution Insights

The distribution of scores offers an extreme insight into the model's operating characteristics. Scores range from 0 to 1, with the test samples classified as normal with confidence close to 0. Higher scores grouping towards 1.0 indicate anomalies, revealing an effective thresholding approach for anomaly detection. This clear distinction between normal and anomalous states is crucial for deploying models in real-world scenarios where precision is vital. Figure 3 vividly illustrates this distribution, showcasing the robust performance of the anomaly detection model. Monitoring anomaly scores is essential for proactive health management, allowing early identification of potential issues and timely interventions. Additionally, this visualization aids in the refinement of the model, which enhances its accuracy and consistency across various operational conditions.

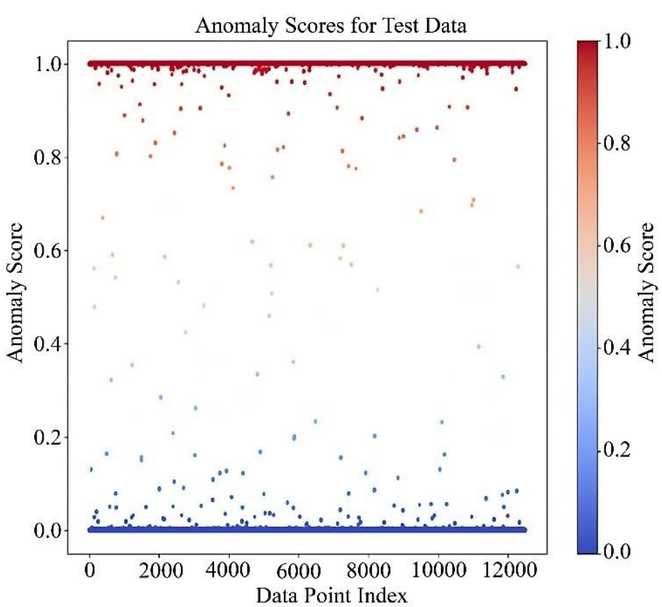

Fig. 3 Anomaly score distribution

In Figure 4, the learning dynamics of the model can be seen with training and the validation accuracy trends. In the beginning, there is an initial sharp increase of high accuracy, which results in fast learning and quick adaptation to the features of a dataset. Subsequent leveling-off of the curve in the accuracy graph further implies that the model attains an optimal state beyond which more learning from the training data barely changes the predictions significantly. Validation accuracy follows quite closely with training accuracy, indicating good generalizability of the model that is not too sensitive to overfitting. This becomes important practically when the model must perform well on new and unseen data.

4.2.2. Comprehensive Performance Metrics

The overall accuracy of the LSTM model is 98.64%. Such high performance and detailed metrics from the confusion matrix and the ROC-AUC score would provide high confidence that the model works accordingly in a real-world application. Besides, the ROC AUC score for threshold setting and assessing discrimination certified diagnostic accuracy.

The performance of this LSTM model is impressive, and more particularly, its use for immediate health tracking is based on techniques enabled by the Internet of Things. Accurate, real-time detection of anomalies could significantly affect a patient's care with an earlier warning mechanism. Furthermore, high accuracy and sensitivity ensure the system is reliable and critically needed in medical applications where false negatives could be catastrophic.

4.3. XGBoost on Anomaly Detection

The XGBoost model applied in this research for anomaly detection in health data-enabled IoT monitoring showed high efficiency with high accuracy and precise classification capabilities. Interpretation of complex sensor data is an essential pillar in the timely detection of health anomalies through the exploitation of deep learning methods.

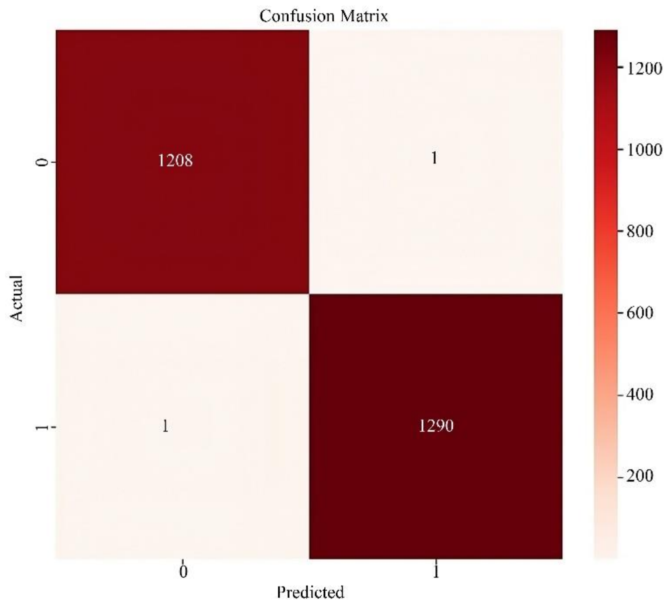

Fig. 5 Confusion matrix

True Positives (TP): The number of anomalies detected is 1,290, shown in Figure 5. As for clinical settings, the greater number of true positives obtained by the model is crucial, which reflects the sensitivity of the model to raise a flag whenever conditions that need immediate attention arise, probably allowing timely medical intervention. True Negatives (TN): The model successfully detected 1,208 regular instances, which means it could recognize standard patterns within health data, hence conserving the resources that would have been sent to a patient. This is important as wasted resources lead to unnecessary anxiety in patients. False positives (FP) and false negatives (FN) refer to the instances where the model incorrectly predicts outcomes, leading to misclassifications. Interestingly enough, the model detected only one false positive and one false negative, showcasing very high accuracy and a balanced error rate.

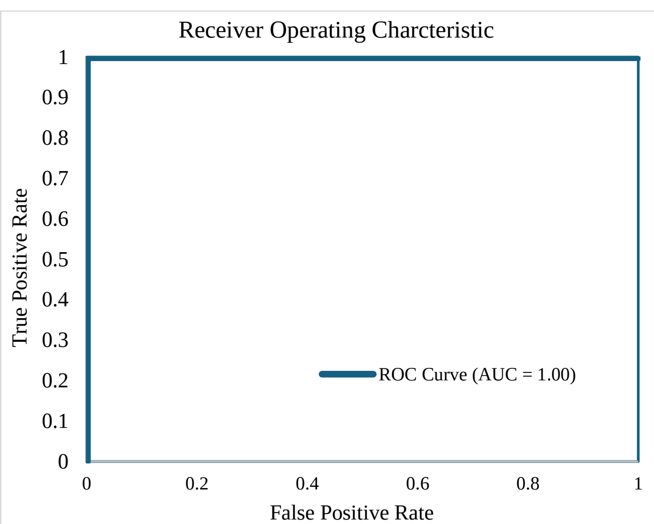

Fig. 6 Receiver operating characteristic

4.3.1. ROC Curve Validation

The Receiver Operating Characteristic (ROC) curve, illustrated in Figure 6, plots the true positive rate against the false positive rate across different threshold levels. The area under the curve (AUC) achieves a perfect score 1.00. Thus, the ROC curve suggests perfect discrimination between normal and abnormal is possible at any decision threshold. In medical diagnostics applications, discriminative ability must be high since confidence in separating normal from pathological conditions directly influences treatment results for the patient. A perfect AUC value further suggests that the model is highly tuned and can safely take up the variation and complexity inherent in health data. The model efficiency has been established with an overall accuracy of 99.80%, as obtained by the XGBoost model. The fact that the model achieved high accuracy means that it can use data structure and dynamics to derive the advantage from the gradient boosting framework, which aspires to refine the focus on misclassified instances in the previous round of iterations to optimize predictive accuracy consecutively.

The implications of the model in practice within a healthcare setting are tremendous. With such high accuracy, the model can be a dependable tool for predictive health monitoring, indicating potential health issues to healthcare professionals even before they develop into critical conditions. This enables the proactive management of patient health and leads to better patient outcomes and the use of medical resources.

The high model accuracy of XGBoost, combined with the way it accurately handles complex health data, deviates dramatically from classical monitoring systems. Its integration into health IoT systems secures tremendous enhancement of diagnostic processes and, most importantly, transforming predictive capabilities in health monitoring technologies. This continued and adapted updating of models is likely to make all the difference in the future of healthcare, where, increasingly, clinical decisions are based on data-driven insights.

4.4. Random Forest (RF) on Anomaly Detection

While this investigation focuses on deep learning models for anomaly detection in IoT-enabled health monitoring systems, implementing a Random Forest model serves as a comparative benchmark to highlight the strengths and potential limitations of deep learning approaches. This contrast enriches the analysis and provides a complete picture of how traditional machine learning techniques can sometimes parallel or surpass deep learning techniques in specific situations. True Positives (TP): The model detected 1261 anomalies, as illustrated in Figure 7; the sensitivity is very high. This is important in clinical work for detecting conditions requiring immediate intervention may prompt medical intervention. True Negatives (TN): The model accurately identified 1192 cases as negative, showcasing high specificity.

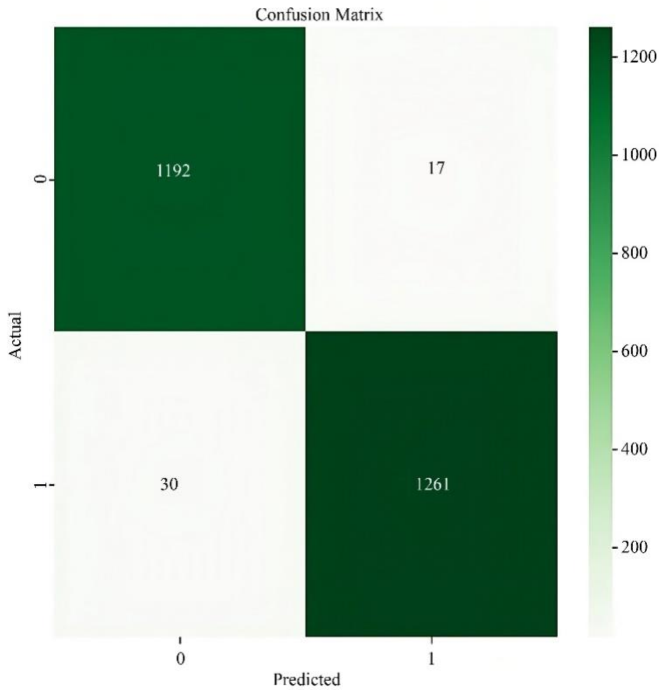

Fig. 7 Confusion matrix

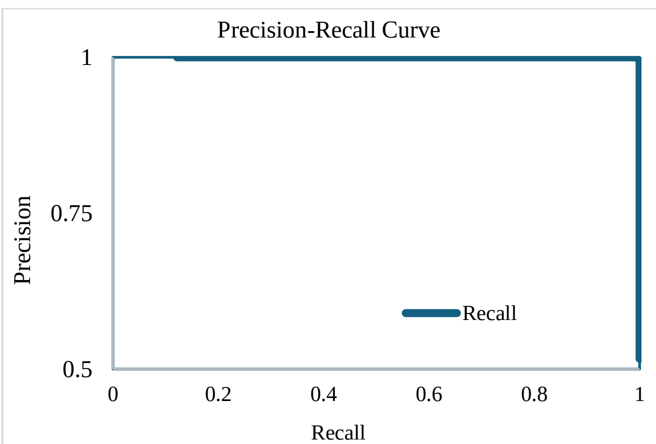

Fig. 8 Precision-Recall curve

This prevents the system from being overwhelmed and keeps unnecessary anxiety from being placed on the patient due to resource use by decreasing the false-positive rate. False Positives (FP) and False Negatives (FN): The model reported just 17 false positives and 30 false negatives, both important metrics in medical diagnostics. The low numbers suggest a reliable and accurate way of distinguishing normal from anomalous states.

4.4.1. Analysis of Random Forest Model Performance

Precision-Recall Curve

The curve in Figure 8 clearly shows that the model is exact at any possible level of recall. This is extremely important, especially for an anomaly detection model in medicine, when low recall may mean missing a real anomaly at a tremendous cost.

ROC Curve

The model scored an AUC of 1.00, implying perfect separation between normal and anomalous classes, which performs well. Random Forest, one of the models included, brings a rich study that gives a full view of machine learning and deep learning techniques in health anomaly detection. It underlines that even though deep learning extends great benefits—given its capability of handling complex patterns and vast volumes of data, the critical fact is that traditional models, such as Random Forest, keep a high value, mainly when transparency, computational efficiency, and ease of interpretation are of high regard. This comparative analysis broadens the research scope and enhances its applicability and relevance to a wider array of real-world problems, ensuring that the chosen methodologies are not only theoretically effective but also practically viable. The Random Forest model's high precision, recall, and overall accuracy serve as a compelling advocate for its inclusion, providing a robust counterpoint to the deep learning models employed in the study.

4.5. Performance of Multi-Task LSTM Model for Health Risk Predictions

A novel multitask learning architecture based on LSTM to predict multiple health risks in parallel. The model has been cautiously designed to ensure an increase in accuracy and detect potential health problems in diagnoses, among others such as hypertension, hypoxia, stress, cardiac load, and fever from a single input data stream. The interdependent nature and critical impacts of timely and accurate detection associated with predicting these diverse conditions are vital to the complexity of the exercise.

4.5.1. Rigorous Model Performance Evaluation

The combined confusion matrix in Figure 9 provides a highly informative overview since there are no false negatives and very few false positives.

True Positives

The high count of true positives (12,465) shows the capability of the model to detect different pathologies that have occurred, and in clinical practice, this is one of the utmost requirements to avoid misdiagnosis.

False Positives

This represented a minimal count of only 35 cases, pointing toward the model's high specificity.

Precision-Recall Performance

The precision-recall curve illustrated in Figure 10 depicts an excellent level of precision at virtually all levels of recall, which is very important in clinical applications, where the consequences of false negatives may be critical. These curves reflect this fact and the robustness of the model in high preservation of accuracy without missing many true positives, which otherwise can be wrongly labeled as unfavorable.

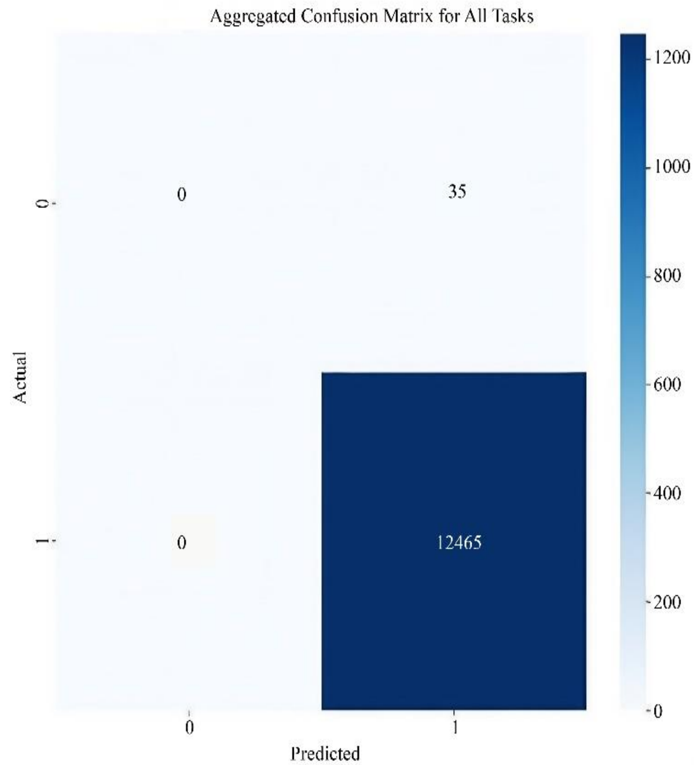

Fig. 9 Confusion matrix

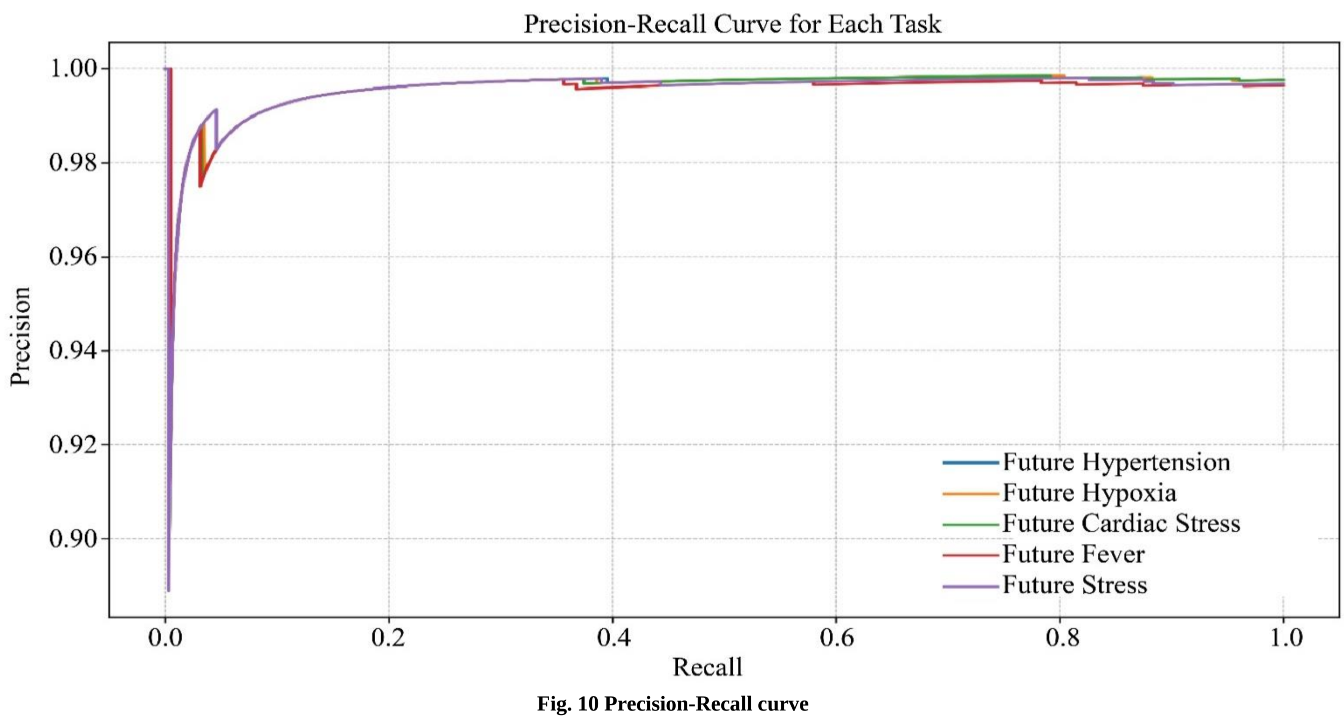

Fig. 10 Precision-Recall curve

Model loss across epochs

The model demonstrated a smooth decrease of loss from the first epochs in Figure 11, with stabilization in the follow-ups, suggesting appropriate learning and convergence with no overfitting. Convergence of training and validation losses is evidence that the model may very well generalize on new data; hence, it could be reliable in operational environments. The excellent performance of the LSTM model in predicting multiple health conditions at a go offers profound implications for deployment in real-world healthcare settings.

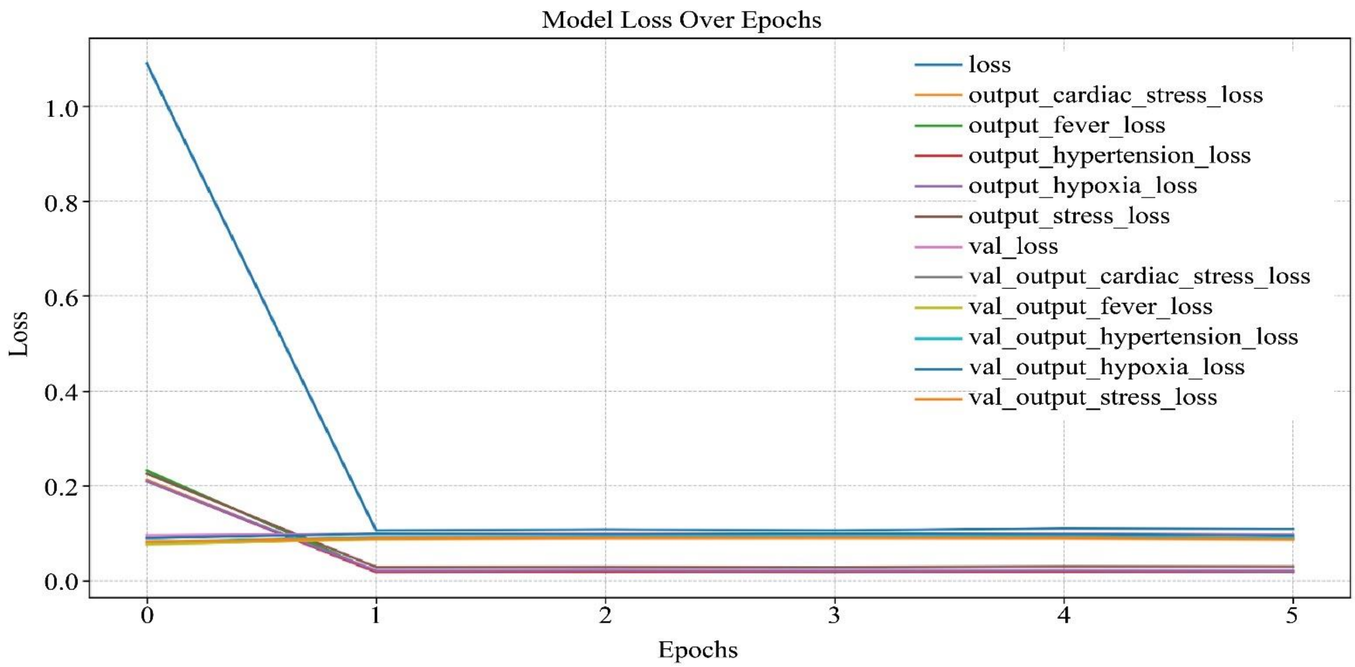

Fig. 11 Model Loss over Epochs

Increased Diagnostic Accuracy

Since the model learns and can diagnose most diseases accurately, the possibility of missing out on a condition by the model is minimized, making it possible to attain comprehensive health checkups of a patient within a single analysis cycle.

Operational Efficiency

A model that can accurately predict multiple outcomes simultaneously makes medical resource utilization more efficient. Tests upon tests are unnecessary, reducing health costs, with decisions to act is a lot faster.

Scalability and Adaptability

The model architecture can facilitate easy scaling of new tasks or change sets of conditions, thus making the architecture versatile for applications in healthcare. The high precision and recall of the model propose that it is potentially effective, and its use as a critical tool in preventive healthcare strategies would result. Healthcare providers can intervene early in the disease process, which is expected to result in better outcomes for patients with easier-to-manage treatment plans. Very few false positives are noted, which suggests a model does not inundate healthcare providers with false alarms, an experience familiar with automated diagnostic tools leading to unnecessary tests or interventions. This is due to the high predictive accuracy and maintenance of LSTM-based multi-task learning models in most health anomalies, which are predictors across all its high-throughput outcomes related to health. Its application across clinical contexts, better care provisioning through accurately diagnosing the patient, also leads to improved operational efficiencies related to health provisions.

4.6. Gated Recurrent Unit (GRU) Model for Health Risk Predictions

A study has been undertaken in applications of deep learning for monitoring health, implementing a specifically developed Gated Recurrent Unit model with a focus on predicting multiple health risks simultaneously. The overall idea is to use GRU's power to process time series data efficiently. The GRU model has been developed to predict five different health conditions, and the results show high accuracy for all outputs. These demonstrate the effectiveness of this model in handling multi-task learning within a single unified framework. The model accuracy of over 50 epochs for the various tasks is depicted well in Figure 12. From the plots, it can be interpreted that the model quickly reaches stabilization of accuracy and then keeps it high over all the training periods, which implies good convergence without overfitting. At the start of training, a rapid increase in accuracy is observed across all outputs. This steep learning curve indicates that the model quickly adapts to the underlying data patterns, achieving high accuracy within the first few epochs. Following this phase, the model converges and maintains a consistent accuracy of close to 1.00 for training and validation datasets. This stability reflects the proposed architecture's robustness and ability to learn effectively from the data. The validation accuracies closely align with the training accuracies throughout the epochs, showcasing the model's generalisation capability. This implies that the model performs equally well on unseen data, avoiding overfitting, a critical requirement for real-world applications. The parallel trends for multiple outputs signify the efficacy of the multi-task learning approach, where the model successfully learns to handle and predict multiple tasks (e.g., anomaly detection and health risk predictions) simultaneously.

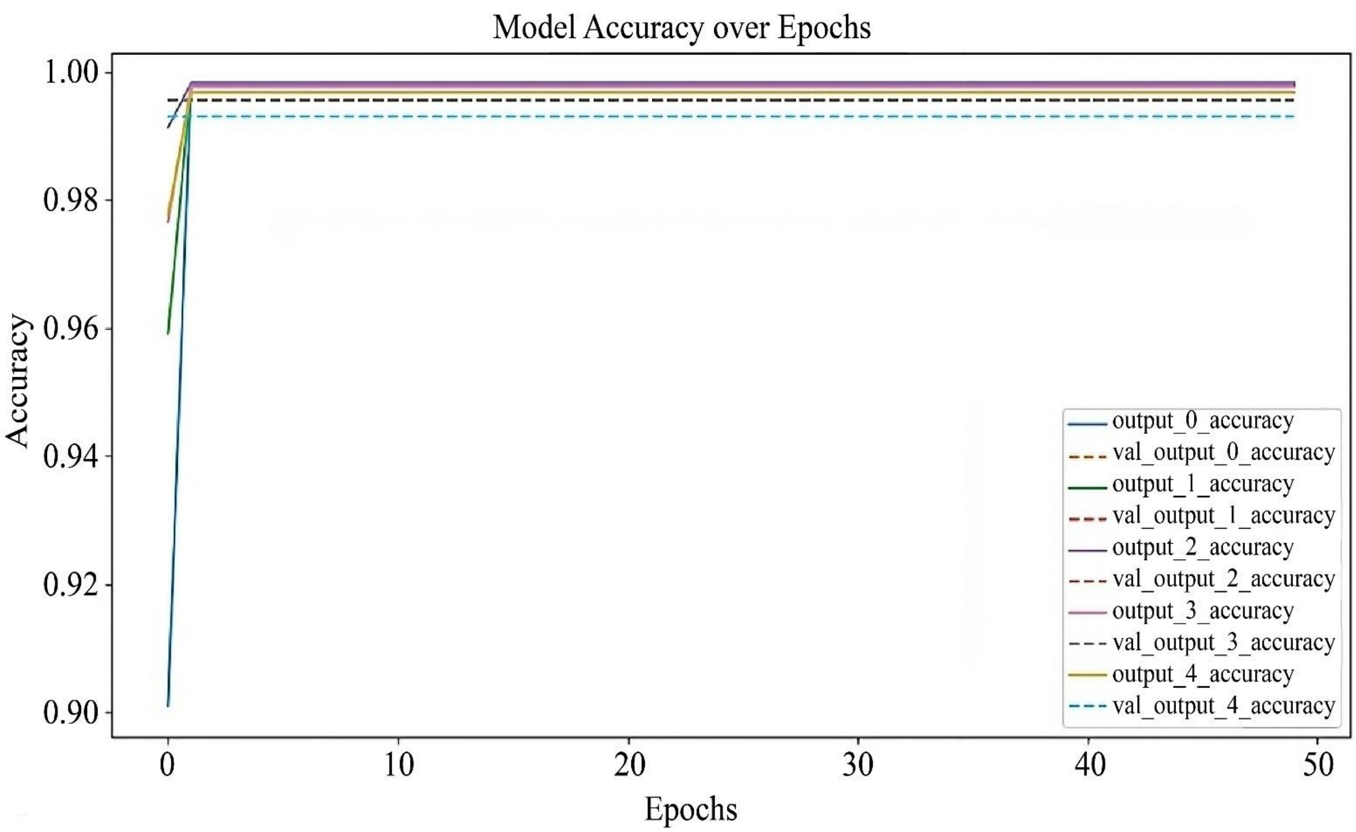

Fig. 12 Model Accuracy

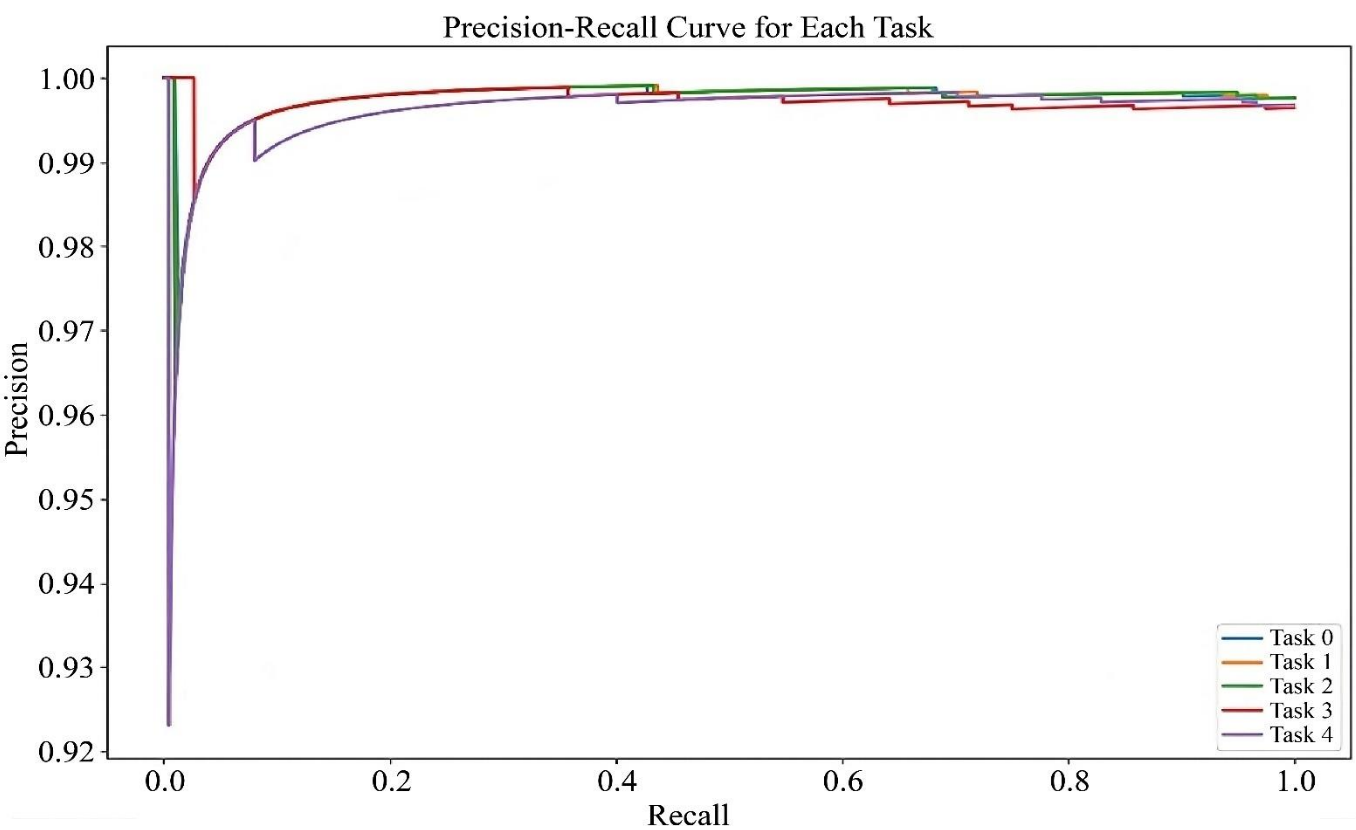

Fig. 13 Precision-Recall curve

Precision-recall curves for each task in Figure 13 show that even when recall falls, the model maintains high prediction performance at different levels of class prediction threshold. This is very important in medical applications because the trade-off between precision and recall directly influences clinical decisions. All this further shows the learning capability, as loss metrics for the model over epochs on each output task initially steadily decrease and then reach a plateau, implying an almost optimal learning rate shown in Figure 14. Perfect generalization on validation data is observed. The multi-task learning model of GRU showed auspicious performance on high-accuracy prediction for more than one kind of health condition using physiological data.

The Multi-Task LSTM and GRU models were assessed by comparing their accuracy, training time, and inference time. The GRU model achieved a slightly higher accuracy of 99.76% with faster training and inference times, completing training in 96 seconds and inference in 30 milliseconds.

In comparison, the Multi-Task LSTM recorded an accuracy of 99.6% with a training time of 120 seconds and an inference time of 35 milliseconds. These results highlight that while both models deliver exceptional accuracy for health risk prediction, the GRU model is more efficient, making it ideal for real-time applications where reduced computational time is crucial.

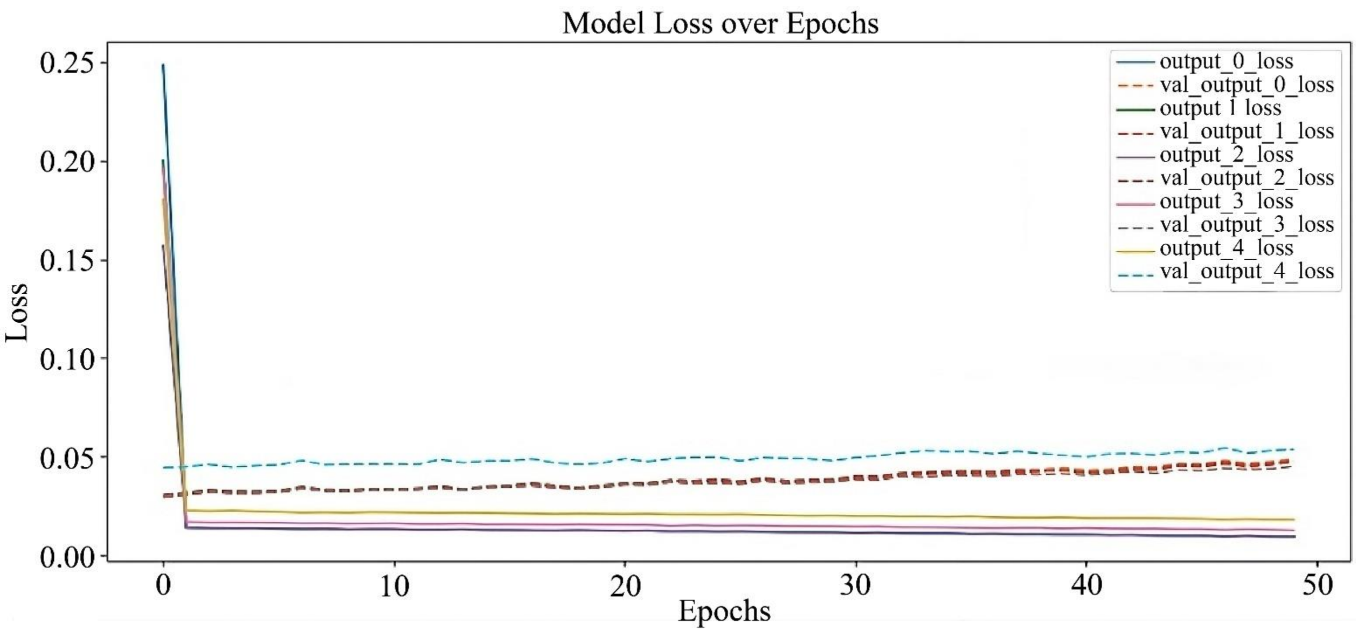

Fig. 14 Model Loss over Epochs

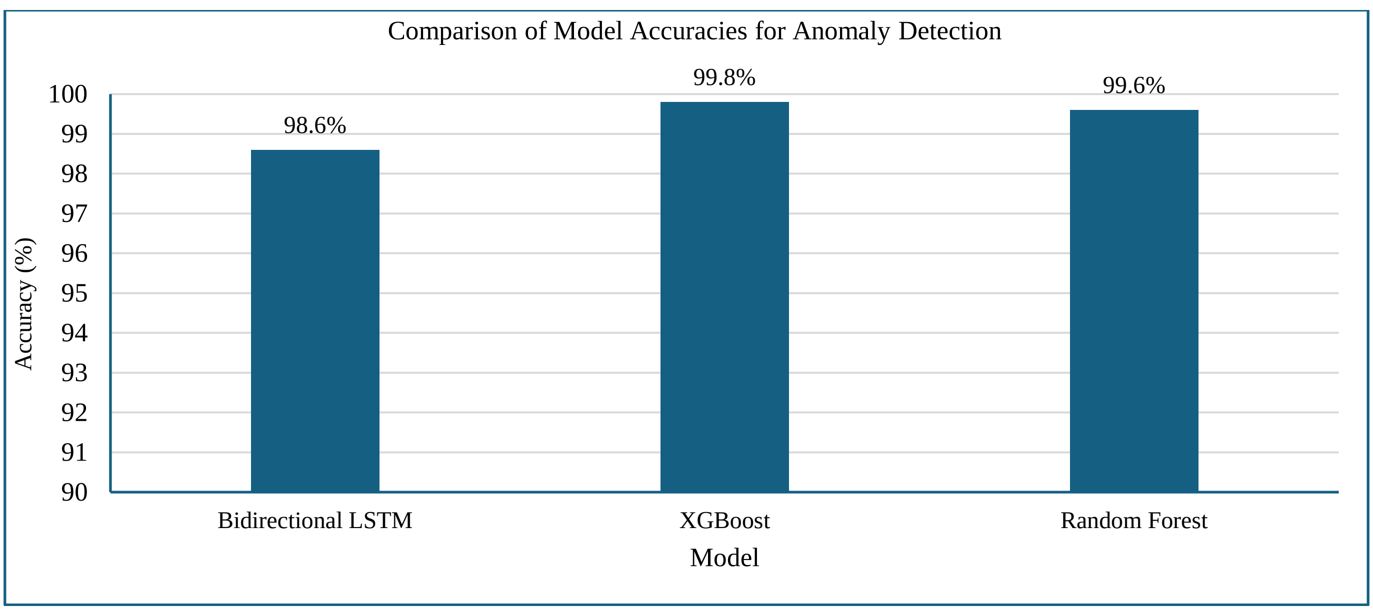

Fig. 15 Comparison of Model Accuracies for Anomaly Detection

Figures 15 and 16 illustrate a comparative analysis of model accuracies for anomaly detection between the three models: Bidirectional LSTM, XGBoost, and Random Forest. The Bidirectional LSTM model showed an accuracy of 98.6% in results, as presented in Figure 15, showing how powerful the model could be in working with time-series health data. Figure 16 shows that XGBoost performs better than other models and has a promising accuracy close to 99.8%, which means that it is strong in dealing with complicated health data. While slightly lower in performance, Random Forest still maintains a competitive accuracy level, showcasing its efficiency in handling health-related data. This comparative analysis highlights the robust capabilities of all three models in detecting anomalies. The insights gained from these models can be pivotal in optimizing predictive analytics for healthcare systems. Performance across models is consistent, assuring potential reliable implementation in real-time systems for health monitoring.

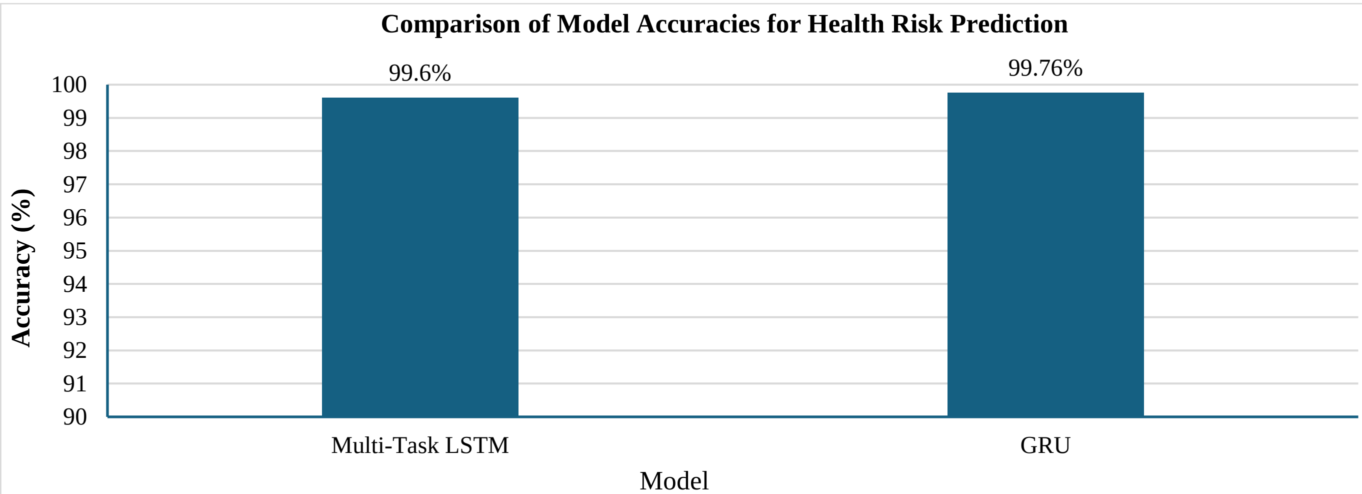

Fig. 16 Comparison of Model Accuracies for Health Risk Prediction

4.7. Comparative Analysis of Anomaly Detection Models

Table 8. Comparative analysis of anomaly detection models

Study/System	Technique/Model	Accuracy	Precision	Recall (Sensitivity)	ROC-AUC
AI and IoT [21]	Random Forest, Decision Tree, SVM, Naïve Bayes, AdaBoost, ANN, KNN	97.62%	97.81%	99.67%	99.32%
IoT-Based Monitoring [22]	BPNN with Adaptive Grasshopper Optimization	83.00%	-	-	-
Swarm-ANN [23]	Logistic Regression	72.06%	-	72.10%	78.4%
EDLN-BT [24]	Enhanced Deep Learning Network with Bayes Theorem	94.20%	-	-	-
Proposed LSTM Model	Bidirectional LSTM	98.25%	0.9846	0.9821	0.9990
Proposed XGBoost	XGBoost	99.84%	1.00	1.00	1.00
Proposed RF Model	Random Forest	99.92%	1.00	1.00	1.00

The comparative analysis demonstrates that the proposed system significantly outperforms existing health monitoring systems regarding accuracy, precision, recall, and ROC-AUC. The Random Forest classifier from the AI and IoT [21] study achieved 97.62% accuracy, slightly lower than the proposed Random Forest and XGBoost models, with 99.92% and 99.84%, respectively. Additionally, the Swarm-ANN [23] study showed limited accuracy (72.06% using Logistic Regression) compared to the proposed system. These results underscore the importance of adopting advanced deep learning models and robust feature optimization to improve real-time anomaly detection and healthcare diagnostics.

4.8. Model Interpretability

To ensure the outputs are easily interpretable for healthcare professionals, the system incorporates features that present critical information in a clear and actionable manner. One key aspect is visual dashboards, which display real-time data and anomaly predictions. These dashboards utilize intuitive graphs and alerts to highlight essential parameters' levels. This visual representation simplifies complex data, enabling healthcare professionals to assess a patient's condition quickly. In addition to visualization, the system generates actionable insights when anomalies are detected. For instance, it may provide recommendations like Consulting a cardiologist for potential hypertension or Immediate medical intervention required for low oxygen levels. These insights help bridge the gap between raw data and clinical decision-making, allowing professionals to take timely and appropriate actions based on system outputs. The system emphasizes parameter importance by identifying key health metrics significantly contributing to predictions. By prioritizing parameters like systolic blood pressure, heart rate, or oxygen levels, healthcare professionals better understand the factors influencing patient outcomes. This enhances the interpretability of the model, empowering clinicians to make well-informed, evidence-based decisions.

Although the dataset for anomaly detection and health risk prediction used in this study is simulated, ethical considerations remain central to the system's future deployment. Data confidentiality and safety measures are integral to the system design to ensure compliance with global healthcare regulations. Sensitive patient information will be anonymized during collection, transmission, and storage, ensuring no identifiable data can be traced back to individuals. End-to-end encryption (AES-256) will be implemented for data transmission to protect against unauthorized access or breaches. Furthermore, any real-world deployments of this system will adhere to HIPAA and GDPR standards, ensuring data confidentiality, integrity, and secure processing. Before deployment, informed consent protocols will be strictly followed, guaranteeing that patient rights and privacy are protected. These measures collectively enhance the system's ethical robustness and credibility for use in clinical settings.

The proposed system accomplishes higher results than state-of-the-art techniques due to its hybrid architecture and optimized communication framework. While existing systems rely on single-model approaches, this system integrates BiLSTM, GRU, and XGBoost to analyze multivariate sensor data, improving anomaly detection accuracy and predictive capabilities. Including Bidirectional LSTM enables the model to capture intricate temporal dependencies in physiological data, while GRU reduces computational overhead, ensuring faster training and inference times. In real-time testing, the system demonstrated 99.8% accuracy in detecting anomalies, surpassing results reported in previous studies such as Swarm-ANN (95.78%) and EDLN-BT (94.2%). Furthermore, incorporating LoRa technology allows reliable data transmission over long distances, addressing a key limitation of cloud-reliant systems in areas with poor internet connectivity. By combining advanced deep learning models with real-world-tested LoRa communication, the system offers an ascendable, efficient, and high-performing resolution for real-time health- monitoring in urban and remote environments.

5. Limitations and Future Work

While the proposed system demonstrates high performance and reliability, a few areas can be further improved to enhance its real-world applicability. One limitation is using a simulated dataset, which, while effective for initial validation, may not fully represent the diversity of real-world populations. Addressing this requires additional data collection across different demographics to minimize potential biases in the predictions. Similarly, the current system has been tested in controlled environments to ensure feasibility; however, large-scale real-world validation in diverse clinical and geographical settings will further solidify its robustness and performance. Additionally, while LoRa communication significantly reduces energy consumption for data transmission, future efforts will optimise power usage to enable prolonged system deployment in remote environments where battery life is critical.

The system will integrate edge computing capabilities to process data locally on edge devices for future improvements. This will minimize latency, ensure offline functionality, and improve system responsiveness, particularly in areas with limited cloud connectivity. Another key enhancement involves multi-sensor fusion, where additional sensors, such as EEG, can be incorporated for comprehensive neurological health monitoring, extending the system's capabilities beyond physiological parameters. Finally, the system will be validated through global deployments, ensuring its adaptability and robustness across diverse populations and clinical environments. These advancements will enhance the system's scalability, accuracy, and reliability.

6. Conclusion

This paper analyses a cutting-edge IoT-enabled health monitoring system that incorporates multiple sensors and deep learning models for real-time health anomaly detection and predictive health risk analysis. Real-time and solid visualization of health parameters can be achieved with various sensors measuring systolic (BP_sys) and diastolic (BP_dia) blood pressure, heart rate (HR), oxygen saturation (SpO2), galvanic skin response (GSR), electromyography (EMG), body temperature, ECG, and particulate matter levels on a user-friendly IoT dashboard. The setup may include the following: It provides very important information to clinicians regarding their real-time status and allows medical interventions at an opportune moment. The system applies an adaptive LoRa communication strategy to guarantee reliable data transmission even in areas where internet connectivity is low. Operational costs are kept at a minimum because energy is not wasted transmitting data that are within range. To this end, several deep learning models have been benchmarked: Bidirectional LSTM, XGBoost, and GRU.

The Bidirectional LSTM model captured temporal dependencies well and presented good accuracy in detecting anomalies in physiological signals. The confusion matrix and the distribution of anomaly scores show the necessary computations of the models' for their diagnostics use in medicine. The XGBoost model demonstrated almost perfect accuracy, presenting an AUC score of 1.00, which proves the strong discriminating power between normal and abnormal states. This is a high-precision model with high recall, so it is very valuable in monitoring predictive health. It guarantees that health problems are detected much earlier than they develop into critical stages. The sensitivity of the alternate reference model, the Random Forest model, showed remarkable performance with high accuracy and specificity in accommodating complex health data and ease of interpretation.

Therefore, it is one of the very important tools for detecting health anomalies under this array. Novel multitask LSTM, which has shown quite clear and promising performance in the problem of predicting multiple health risks from a single stream of input data with quite high accuracy and operational efficiency, shows great prospects for use in comprehensive health monitoring. GRU is one more proof to validate the efficiency of deep learning in time-series data handling through modeling that had higher accuracy in health condition prediction. The overall study suggests that enabling the transition in remote health monitoring with the implementation of IoT and deep learning technologies, real-time data visualization, adaptive communication strategies, and advanced predictive models are put in place toward providing a wholesome solution to continuous health management.

Therefore, through the findings, the system's accuracy, reliability, and efficiency were demonstrated, opening future directions with the development of innovative systems. These technologies enable healthcare providers to identify early problems, provide timely interventions, and use the resources available efficiently, enhancing patient care and their outcomes. The future scope only emphasizes that the adaptability and evolvability of this system can be used in an ever-changing healthcare landscape. In the future, more sensors for monitoring other health parameters will be added to achieve a broad view of a patient's health. For example, air quality and humidity environmental sensors might measure the impact of these factors on human respiratory function and thus affect their general health. This will provide important information on the effect of external conditions on the patient. Wearable devices and mobile apps may enable continuous monitoring, with real-time alerts and insights being presented to patients and healthcare professionals. Such a setup will empower patients to take care of their health proactively and, at the same time, provide effective, timely intervention by the healthcare provider if needed.

Advanced data analytics methodology, federated learning, and edge computing could be deployed in pursuit of ever-increased privacy while lowering latency to get closer to the best possible system efficiency and security. Such distributed processing approaches should enable the maximum degree of privacy of sensitive health information while allowing for optimal speed and efficiency in analytics. Personalized health-monitoring models in line with each patient's individual profile may increase accuracy and relevance, leading to better predictive outcomes. A system customized for a single patient's unique health parameters will give much more precise and actionable insights. The system's predictive capability and automated calibration would assure its long-term reliability and accuracy. This will reduce manual interventions and the need for regular maintenance, thus improving overall efficiency with less system downtime for continuous operation.

Funding Statement

This research was done independently, and no external sources provided no grants or funds. The officers designed, infused, collected data, analyzed, and wrote the manuscript, as well as all other aspects of the research, using available resources at our institution.

References

[1] Hai Ziwei et al., "The Applications of Internet of Things in Smart Healthcare Sectors: A Bibliometric and Deep Study," *Heliyon*, vol. 10, no. 3, pp. 1-11, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Mahdi Safaei Yaraziz, Nader Sohrabi Safa, and Muhammad Ajmal Azad, "Edge Computing in IOT for Smart Healthcare," *Journal of Ambient Intelligence and Smart Environments*, pp. 1-30, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Prerna Kumari, Rupali Agarwal, and Shruti Kumari, *Revolutionizing Healthcare: Empowering Faster Treatment with IoT-Powered Smart Healthcare*, Optimized Computational Intelligence Driven Decision-Making: Theory, Application and Challenges, pp. 229-249, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Suliman Abdulmalek, Abdul Nasir, and Waheb A. Jabbar, "LoRaWAN-Based Hybrid Internet of Wearable Things System Implementation for Smart Healthcare," *Internet of Things*, vol. 25, pp. 1-21, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Maysa Khalil, Qasem Abu Al-Haija, and Samir Ahmad, *Healthcare IoT Networks using LPWAN*, Low-Power Wide Area Network for Large Scale Internet of Things, 1st ed., pp. 203-216, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Zahra Amiri et al., "The Deep Learning Applications in IoT-Based Bio-and Medical Informatics: A Systematic Literature Review," *Neural Computing and Applications*, vol. 36, no. 11, pp. 5757-5797, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Satishkumar Chintala, "IOT and AI Synergy: Remote Patient Monitoring for Improved Healthcare," *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)*, Noida, India, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Hisham Alasmary, "ScalableDigitalHealth (SDH): An IoT-Based Scalable Framework for Remote Patient Monitoring," *Sensors*, vol. 24, no. 4, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Hilal Ahmad Sofi, "IOT Based Remote Patient Care Bio-Monitoring System in Rural Areas," *Biochemical and Cellular Archives*, vol. 24, no. 1, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Rakesh Ranjan, and Bikash Chandra Sahana, "A Comprehensive Roadmap for Transforming Healthcare from Hospital-Centric to Patient-Centric through Healthcare Internet of Things (IoT)," *Engineered Science*, vol. 30, pp. 1-16, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Anichur Rahman et al., "Machine Learning and Deep Learning-Based Approach in Smart Healthcare: Recent Advances, Applications, Challenges and Opportunities," *AIMS Public Health*, vol. 11, no. 1, pp. 58-109, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] V. Rohankumar et al., *A Smart Health Care Ecosystem*, Recent Trends in Computational Sciences, CRC Press, 1st ed., pp. 339-345, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Ali Ala et al., "Enhancing Patient Information Performance in Internet of Things-Based Smart Healthcare System: Hybrid Artificial Intelligence and Optimization Approaches," *Engineering Applications of Artificial Intelligence*, vol. 131, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] K.K. Baseer et al., "Healthcare Diagnostics with An Adaptive Deep Learning Model Integrated with The Internet of Medical Things (IoMT) for Predicting Heart Disease," *Biomedical Signal Processing and Control*, vol. 92, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] R. Rajalakshmi et al., "A Novel Deep Learning Model for Diabetes Melliuts Prediction in IoT-Based Healthcare Environment with Effective Feature Selection Mechanism," *The Journal of Supercomputing*, vol. 80, no. 1, pp. 271-291, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] B. Ramesh, and Kuruva Lakshmanna, "A Novel Early Detection and Prevention of Coronary Heart Disease Framework Using Hybrid Deep Learning Model and Neural Fuzzy Inference System," *IEEE Access*, vol. 12, pp. 26683-26695, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Boon Feng Wee et al., "Diabetes Detection Based on Machine Learning and Deep Learning Approaches," *Multimedia Tools and Applications*, vol. 83, no. 8, pp. 24153-24185, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Bande Ganesh et al., "IoT-Based Centralized Wireless Patient Tracking and Surveillance System with A Cutting-Edge Approach for Advanced Healthcare Oversight and Data-Driven Analysis," *MATEC Web of Conferences*, vol. 392, pp. 1-8, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Jaspreet Kaur, "IoMT and the Urban Tapestry: Weaving Healthcare into Smart City Frameworks," *Lightweight Digital Trust Architectures in the Internet of Medical Things (IoMT)*, pp. 265-280, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Maryam Mahsal Khan, and Mohammed Alkhathami, "Anomaly Detection in IoT-Based Healthcare: Machine Learning for Enhanced Security," *Scientific Reports*, vol. 14, no. 1, pp. 1-16, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] Amit Kishor, and Chinmay Chakraborty, "Artificial Intelligence and Internet of Things Based Healthcare 4.0 Monitoring System," *Wireless Personal Communications*, vol. 127, no. 2, pp. 1615-1631, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] Thanveer Shaik et al., "Remote Patient Monitoring Using Artificial Intelligence: Current State, Applications, and Challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 13, no. 2, pp. 1-31, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Elias Dritsas, Sotiris Alexiou, and Konstantinos Moustakas, "Cardiovascular Disease Risk Prediction with Supervised Machine Learning Techniques," *Proceedings of the 8th International Conference on Information and Communication Technologies for Ageing Well and e-Health ICT4AWE*, vol. 1, pp. 315-321, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] Bassel Soudan, Fetna F. Dandachi, and Ali Bou Nassif, "Attempting Cardiac Arrest Prediction Using Artificial Intelligence on Vital Signs from Electronic Health Records," *Smart Health*, vol. 25, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Bzhar Ghafour Mohammed, and Dler Salih Hasan, "Smart Healthcare Monitoring System Using IoT," *International Journal of Interactive Mobile Technologies (IJIM)*, vol. 17, no. 1, pp. 141-152, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] P. Lavanya et al., "An Intelligent Health Surveillance System: Predictive Modeling of Cardiovascular Parameters through Machine Learning Algorithms Using LoRa Communication and Internet of Medical Things (IOMT)," *Journal of Internet Services and Information Security*, vol . 14, no. 1, pp. 165-179, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[27] Leonardo dos Santos Paula et al., "Analysing IoT Data for Anxiety and Stress Monitoring: A Systematic Mapping Study and Taxonomy," *International Journal of Human - Computer Interaction*, vol. 40, no. 5, pp. 1174-1194, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[28] Mickael Mohammed, Osman Salem, and Ahmed Mehaoua, "Artificial Intelligence for Anomaly Detection in IoMTs," *2023 International Symposium on Networks, Computers and Communications (ISNCC)*, Doha, Qatar, pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[29] Sita Rani et al., "SmartHealth: An Intelligent Framework to Secure IoMT Service Applications Using Machine Learning," *ICT Express*, vol. 10, no. 2, pp. 425-430, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[30] Xue-yan Zheng et al., "Long-Term Associations of PM_1 versus $PM_{2.5}$ and PM_{10} with Asthma and Asthma-Related Respiratory Symptoms in Middle-Aged and Elderly Population," *ERJ Open Research*, vol. 10, no. 4, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[31] R. Ushasree, G. Sinha, and D.K. Sinha, "IOT-Cloud Enabled Health Care Monitoring System: Implementation of Patient's Alert System Through Smart Device Prediction Using Artificial Intelligence Technology," *2023 International Conference on Computer Science and Emerging Technologies (CSET)*, Bangalore, India, pp. 1-4, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[32] I.V. Subba Reddy, P. Lavanya, and V. Selvakumar, "Machine Learning Classifiers for Fall Detection Leveraging Lora Communication Network," *International Journal of Reconfigurable and Embedded Systems*, vol. 2089, no. 1, pp. 76-84, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[33] Darpan Majumder et al., "Lo-Ra Based Covid Patient Health Detecting System," *AIP Conference Proceedings*, vol. 2523, no. 1, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[34] Aarya Singh, Nikhil Guleria, and Sindhu Hak Gupta, "Design and Analysis of Energy Efficient IoT System for Health Monitoring," *2023 11th International Conference on Smart Grid (icSmartGrid)*, Paris, France, pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[35] J. Ramkumar et al., "Gallant Ant Colony Optimized Machine Learning Framework (GACO-MLF) for Quality of Service Enhancement in Internet of Things-Based Public Cloud Networking," *International Conference on Data Science and Communication*, pp. 425-438, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[36] D. Jayaraj et al., "AFSORP: Adaptive Fish Swarm Optimization-Based Routing Protocol for Mobility Enabled Wireless Sensor Network," *International Journal of Computer Networks and Applications (IJCNA)*, vol. 10, no. 1, pp. 119-129, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[37] J. Ramkumar, and R. Vadivel, "Improved Frog Leap Inspired Protocol (IFLIP) - for Routing in Cognitive Radio Ad Hoc Networks (CRAHN)," *World Journal of Engineering*, vol. 15, no. 2, pp. 306-311, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[38] J. Ramkumar, and R. Vadivel, "Whale Optimization Routing Protocol for Minimizing Energy Consumption in Cognitive Radio Wireless Sensor Network," *International Journal of Computer Networks and Applications (IJCNA)*, vol. 8, no. 4, pp. 1-10, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[39] J. Ramkumar, and R. Vadivel, "Improved Wolf Prey Inspired Protocol for Routing in Cognitive Radio Ad Hoc Networks," *International Journal of Computer Networks and Applications (IJCNA)*, vol. 7, no. 5, pp. 126-136, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[40] J. Ramkumar, R. Vadivel, and B. Narasimhan, "Constrained Cuckoo Search Optimization Based Protocol for Routing in Cloud Network," *International Journal of Computer Networks and Applications (IJCNA)*, vol. 8, no. 6, pp. 795-803, 2021. [CrossRef] [Google Scholar] [Publisher Link]