



IoT-Based System for Monitoring Diabetes Patients During the COVID-19 Pandemic

Prepared By

Name: Sami Lotfi Almaghrabi

ID No.: 251002918

Submitted in fulfillment of the requirements for the degree Master of
Science in Artificial Intelligence

At

Collage of Informatics

Midocean University

UAE

Supervisor Name: Ashraf Alsayed

Collage of Informatics, Midocean University, Comoros

Acknowledgements

I would like to express my sincere gratitude to Dr. Ashraf Alsayed for his invaluable guidance, continuous support, and encouragement throughout this research project. His expertise, insightful feedback, and unwavering dedication have significantly contributed to the quality and direction of this thesis.

I also extend my heartfelt appreciation to my family, friends, and colleagues, whose patience, motivation, and moral support were instrumental in the successful completion of this work.

Abstract

The rapid deterioration into serious illness that diabetes may cause was brought to light by the COVID-19 pandemic. This made it even more vital to establish good and safe ways to monitor people from a distance. Because of lockdowns and overcrowding in hospitals, established healthcare systems don't always offer high-risk patients the treatment they require. This project suggests a clever Internet of Things (IoT) tracking system that would allow diabetic patients to take care of their own health from home. This would fix the problems that the pandemic caused and make it possible for digital healthcare to occur in the future.

The device gets information from four separate sensors regarding four vital biological processes (blood glucose levels, oxygen saturation, pulse rates, and body temperature) all the time over the internet. Once this multidimensional data is collected, it will be processed locally, on an edge computing device (such as a Raspberry Pi), using advanced algorithms and interpolative techniques to ensure that the data is accurate and properly temporally aligned. Additionally, the same edge device has two hybrid deep learning engines: one will be a one-dimensional convolutional neural network (1D-CNN), and the other will be a long short-term memory (LSTM) network, which will be used together to detect anomalies in the real time, by providing rapid identification of early signs of changing body functions that may lead to significant health complications.

This new technology looks at each patient's data separately, so doctors only get notifications that are relevant to them. There won't be a lot of notifications; only the ones that matter will be sent. Using z scores and ad hoc tests, the system decides what is "normal" for each patient and sets unique limits for finding unusual observations. The System also cares about keeping data private. It is possible for all devices utilizing federated Learning to perform their training locally and to only send encrypted updates (not actual patient data) to a centralized cloud. Lastly, alerts created by the system do not use a Blackbox algorithm to provide an explanation for receiving an alert due to SHAP values being available for every alert made by the system; therefore, providing the physician with a trustworthy source of information on how the system arrived at that alert.

These results show that our system provides accurate predictions, runs in less than 100 milliseconds, and has less false alarms than other rule-based methods. This

IoT-AI device lets you safely and discretely keep an eye on diabetes patients no matter where you are. It can support multiple users at the same time. It does help to radically improve patient safety and provide stability in the healthcare system regardless of whether you are managing diabetes regularly, or managing through a significant healthcare crisis.

Table of Contents

Acknowledgements	ii
Abstract	iii
Table of Contents	iv
List of Abbreviations	viii
List of Figures	x
List of Tables	xi
Chapter 1 Introduction.....	1
1.1 Overview	1
1.2 Problem Definition.....	2
1.3 Thesis Objectives.....	2
1.4 Thesis Contribution.....	2
1.5 Thesis Organization.....	3
1.6 Chapter Summary	4
Chapter 2 Background and Related Work.....	5
2.1 Introduction.....	5
2.2 Background.....	5
2.3 Related Work	7
2.3.1 Artificial Intelligence-Based Wearable Sensing Technologies.....	7
2.3.2 LoRaWAN-Based Monitoring for Quarantined Diabetics.....	8
2.3.3 Hybrid Edge-Cloud AI Gateway with 1D-CNN for Real-Time Anomaly Detection	8
2.3.4 IoT-Based Approach for Diabetes Patient Monitoring Using Machine Learning	9
2.3.5 Blockchain-Integrated Security for Real-Time Patient Monitoring Using Federated Learning.....	9
2.3.6 Meta-Analysis of IoT Chronic Disease Management.....	10
2.3.7 Comparison of NB-IoT and Bluetooth for Rural Monitoring.....	10
2.3.8 Efficient Prediction of Early-Stage Diabetes Using XGBoost with	

Random Forest Feature Selection	11
2.3.9 COVID-19-Induced New-Onset Diabetes: Trends and Technologies	11
2.3.10 Behavioral Change Interventions for Diabetes Management Through IoT Platforms.....	11
2.4 Synthesis of Literature and Identification of Research Gaps.....	12
2.4.1 Methodological Archetypes	12
2.4.2 Identified Gaps	12
2.5 Research Directions	13
2.6 Chapter Summary	14
Chapter 3 Proposed Approach	15
3.1 Introduction.....	15
3.2 Methodology Overview	15
3.3 System Design Objectives	16
3.4 System Architecture Overview	17
3.4.1 Sensor Layer.....	17
3.4.2 Edge Layer	18
3.3.3 Cloud Layer.....	18
3.5 Dataset Description.....	19
3.6 Data Pre-possessing	20
3.6.1 Synchronizing Different Sensors	20
3.6.2 Removing Noise.....	20
3.6.3 Filling Missing Data.....	21
3.6.5 Features Extraction.....	22
3.7 The Machine Learning Models.....	23
3.7.1 Edge Model: Temporal Convolutional Network with Attention.....	23
3.7.2 Personalizing for Each Patient	25
3.7.3 Anomaly Detection: Variational Autoencoder.....	27
3.7.4 Learning from All Patients: Federated Learning.....	29
3.8 Generating and Explaining Alerts.....	29

3.8.1 When to Alert.....	29
3.8.2 Explaining Alerts to Clinicians	31
3.9 Privacy and Security	32
3.9.1 Protecting Data.....	32
3.9.2 Controlling Access.....	33
3.9.3 Privacy by Design.....	33
3.9.4 Regulatory Compliance.....	33
3.10 Testing and Evaluation	33
3.10.1 Test Setup.....	33
3.10.2 Evaluation Metrics	34
3.10.3 Ablation Studies.....	35
3.11 Chapter Summary	36
Chapter 4 Experimental Results and Discussion	37
4.1 Overview.....	37
4.2 Experiments Setup.....	37
4.2.1 Experimental Environment.....	38
4.2.2 Dataset Description	38
4.2.3 Dataset Verification.....	39
4.2.4 Data Splitting and Cross-Validation.....	39
4.2.5 Preprocessing Pipeline	40
4.2.6 Model Configuration.....	40
4.2.7 Evaluation Metrics	41
4.2.8 Experimental Workflow Summary	42
4.3 Results and Performance Analysis.....	43
4.3.1 Edge Model Performance.....	44
4.3.2 XGBoost Cloud Ensemble Results.....	45
4.3.3 Hybrid Meta-Model Results.....	46
4.3.4 Combined Results Summary.....	47
4.3.5 Latency Analysis.....	47

4.3.6 Alert Behavior and Signal Dynamics.....	48
4.3.7 Making Alerts Explainable.....	49
4.3.8 Discussion of Results.....	50
4.4 Robustness Testing.....	51
4.4.1 Cross-Validation Across Patient Populations.....	51
4.4.2 Robustness to Gaussian Noise.....	52
4.4.3 Robustness to Missing Data.....	53
4.4.4 Cross-Condition Generalization.....	55
4.4.5 Discussion of Robustness Findings.....	55
4.5 Model Comparison to Recent Work.....	56
4.6 Limitations and Future Work.....	58
4.7 Chapter Summary.....	59
Chapter 5: Conclusion and Future Work.....	60
5.1 Introduction.....	60
5.2 Summary of Contributions.....	60
5.3 Achievement of Research Objectives.....	61
5.4 Implications for Clinical Practice and Digital Health.....	62
5.5 Limitations and Critical Reflection.....	63
5.6 Future Research Directions.....	64
References.....	65

List of Abbreviations

- **1D-CNN:** One-Dimensional Convolutional Neural Network
- **AI:** Artificial Intelligence
- **ANN:** Artificial Neural Network
- **ANOVA:** Analysis of Variance
- **AUC:** Area Under the Curve
- **AUPRC:** Area Under the Precision-Recall Curve
- **AUROC:** Area Under the Receiver Operating Characteristic Curve
- **BiLSTM:** Bidirectional Long Short-Term Memory
- **BLE:** Bluetooth Low Energy
- **CGM:** Continuous Glucose Monitoring / Continuous Glucose Monitor
- **CNN:** Convolutional Neural Network
- **COVID-19:** Coronavirus Disease 2019
- **ECG:** Electrocardiography
- **EWMA:** Exponentially Weighted Moving Average
- **FL:** Federated Learning
- **GDPR:** General Data Protection Regulation
- **HIPAA:** Health Insurance Portability and Accountability Act
- **HR:** Heart Rate
- **HRV:** Heart Rate Variability
- **IoMT:** Internet of Medical Things
- **IoT:** Internet of Things
- **KNN:** K-Nearest Neighbors
- **LoRaWAN:** Long Range Wide Area Network
- **LSTM:** Long Short-Term Memory
- **ML:** Machine Learning
- **MLR:** Multiple Linear Regression
- **MQTT:** Message Queuing Telemetry Transport
- **NB-IoT:** Narrowband Internet of Things
- **PPG:** Photoplethysmography
- **PRISMA:** Preferred Reporting Items for Systematic Reviews and Meta-Analyses
- **RCT:** Randomized Controlled Trial
- **RF:** Random Forest
- **RNN:** Recurrent Neural Network
- **ROC:** Receiver Operating Characteristic
- **SHAP:** SHapley Additive exPlanations

- **SMBG:** Self-Monitoring of Blood Glucose
- **SMOTE-ENN:** Synthetic Minority Over-sampling Technique - Edited Nearest Neighbors
- **SpO₂:** Blood Oxygen Saturation
- **STD:** Standard Deviation
- **SUS:** System Usability Scale
- **TCN:** Temporal Convolutional Network
- **TLS:** Transport Layer Security
- **VAE:** Variational Autoencoder
- **XAI:** Explainable Artificial Intelligence
- **XGBoost:** Extreme Gradient Boosting

List of Figures

3.1	Edge-Cloud System Architecture for Diabetic Monitoring.....	13
3.2	Kalman Filter Effect on Heart Rate Data.....	17
3.3	Median Filter Effect.....	18
3.4	Sliding Window Extraction.....	18
3.5	EWMA Adaptive Threshold Example.....	23
3.6	VAE Architecture.....	24
3.7	Federated Learning Communication Flow.....	25
3.8	Clinician Alert Dashboard Example.....	27
3.9	Security Controls by Layer.....	28
4.1	Physiological Signal Distributions.....	35
4.2	Experimental Workflow for IoT Diabetic Monitoring Pipeline.....	39
4.3	Edge Model Training History.....	40
4.4	XGBoost Feature Importance.....	41
4.5	Hybrid Meta-Model Performance Visualization.....	42
4.6	Model Performance Comparison.....	43
4.7	Edge Model Inference Latency Distribution.....	44
4.8	Alert Events Over Time.....	45

4.9	SHAP Explanation for Hypoglycemia Alert.....	46
4.10	Cross-Validation Performance by Fold.....	48
4.11	Performance Degradation Under Gaussian Noise.....	49
4.12	Performance Under Missing Data.....	51
4.13	Summary of Robustness Testing.....	53

List of Tables

3.1	Sensors Used in the System.....	15
3.2	Edge Layer.....	16
3.3	Confusion Matrix Structure.....	33
4.1	Experimental Environment Configuration.....	36
4.2	IoT-Based Diabetic Patient Monitoring Dataset Description...	36
4.3	Model Configurations.....	39
4.4	Performance Evaluation Metrics.....	39-40
4.5	Edge TCN with Attention Results.....	42
4.6	XGBoost Ensemble Results.....	43
4.7	Hybrid Meta-Model Results.....	44
4.8	Combined Results Summary.....	45
4.9	C r o s s - V a l i d a t i o n Results.....	49
4.10	Performance Under Increasing Noise Levels.....	50
4.12	Performance Under Missing Data Rates.....	52
4.12	C r o s s - C o n d i t i o n Generalization.....	53
4.13	Comparison with Recent Work.....	54-55

Chapter 1 Introduction

1.1 Overview

The COVID-19 pandemic makes things considerably difficult for persons who had diabetes. It made it tougher for individuals all around the world to seek medical care and put them at a greater risk of being very ill or dying (Kumar et al. 2025). Patients had to make a tough choice: go to clinics and risk becoming sick, or stay home and risk their conditions getting worse, as hospitals were too busy to see them often (Abdulmalek, Nasir, and Jabbar 2024). It became crucial to manage diabetes remotely when it was shown to be associated with greater COVID-19 risks. This is because diabetes may cause consequences such as immune system damage and comorbid health concerns (Warrier and S 2025).

Wearable technology and the Internet of Things (IoT) have facilitated the monitoring of essential health metrics such as blood glucose levels and oxygen saturation (Ayouni et al. 2025). With this system, doctors don't just receive data they get it in real time. That means they can step in right away if they spot anything alarming. Ultimately, this may lead to fewer hospital visits and improved overall health (Khan and Abaoud 2023). Many studies confirm that IoT systems have demonstrated positive results; for example, Shamsabadi et al. (2022) noted that the number of hospital visits decreased significantly, while Krug, Miethe, and Hutschenreuther (2021) reported that diabetic patients were able to identify their symptoms faster than isolated individuals.

The initial implementations of predictive analytics encountered numerous significant issues. Systems that had fixed alert parameters caused an overwhelming number of false alarms that resulted in the breakdown of trust between patients and physicians (Gündoğdu 2023). Additionally, Metwally et al. (2021) claim that sensor errors in everyday usage degrade the quality of the collected data and make it difficult to generate adequate predictive models from that data. Moreover, privacy violations limited access to valuable data potentially useful for developing predictive models of the future; thereby preventing compliance with regulations such as HIPAA and GDPR (Naz et al. 2024).

A fatigue caused by false alerts exists in regard to inaccurate data, with privacy threats further hindering improvements being implemented within systems; these two problems are intertwined (Seltzer et al. 2019) to formulate the approach of implementing an overall methodology using edge computing to process data through the establishment of tailored monitoring thresholds and using advanced methodologies for cleansing data (Rancea et al. 2024). The system proposed would

be reliable and capable of improving patient safety and decreasing response time within a healthcare environment when responding to emergency situations. The results of the study will significantly relieve patient concerns in critical situations and lower the number of diabetes related complications (Shaw et al., 2010).

1.2 Problem Definition

During COVID-19, remote monitoring devices for diabetes patients encountered three significant issues that diminished their utility. First, using set rules for alerts based on standard medical data sometimes didn't take into consideration the distinctive needs of each patient, which led to wrong diagnoses and late notifications. In the second place, the quality of the data that was collected from sensors was hampered by problems with connection and interference, which led to incorrect readings that might mask actual health concerns. Last but not least, the requirement of sharing sensitive health information in order to improve forecasts led to patients' mistrust of monitoring systems and their unwillingness to use them. This resulted in privacy concerns. These problems are brought to light by research, which reveals the need of flexible systems, greater data dependability, and enhanced privacy safeguards in order to maximize the effectiveness of remote patient monitoring capabilities.

1.3 Thesis Objectives

This study seeks to develop and evaluate an IoT-based remote monitoring system that addresses the aforementioned issues. During and after the COVID-19 epidemic, the system will support people with diabetes by sending them more reliable notifications that suit their regular patterns, cleaning up sensor data to make it more accurate, and preserving their privacy via smart design.

- To build a real-time monitoring system that works at the edge.
- To design personalized alert thresholds that adapt to each patient.
- To build data cleaning pipelines that handle real-world sensor problems.
- To test the complete system against existing approaches.

1.4 Thesis Contribution

This thesis offers a number of important contributions to the domains of diabetes care and digital health. These contributions are 1) First, it shows that a new IoT-based monitoring system greatly improves patient safety and independence. Through the provision of dependable notifications, the technology aids diabetes patients in staying securely at home. Both their quality of life and their risk of hospital-acquired illnesses are improved by this, 2) The second

way it improves healthcare efficiency is by giving doctors a better way to prioritize their patients and provide money to the ones who need it the most. Because of this improvement, providers' operations are made easier, and they feel less stress, 3) Third, the findings indicate that healthcare professionals may generate substantial income and save considerable time. The system facilitates home health management, hence enhancing hospitals' resource use. This optimizes expensive hospital bed availability for emergencies and reduces the expenses associated with unnecessary clinic visits, 4) Fourth, this study fills important gaps in the current literature by looking at IoT-based remote monitoring in real-world clinical settings with real patients. It also takes a close look at the often-overlooked issue of patient privacy, and finally, it creates a scalable and dependable monitoring system that makes the healthcare system more resilient and gives us a tool that we can use to better handle patient loads during future public health crises.

1.5 Thesis Organization

The purpose of this thesis is to provide a thorough examination of an Internet of Things-based diabetes monitoring system throughout the duration of the COVID-19 epidemic using five chapters that are interrelated and build from one another through their progression.

Chapter 1: Introduction presents the research findings and establishes the reasons for individuals with diabetes being more vulnerable due to the COVID-19 pandemic and the inadequacies of traditional healthcare systems. In this chapter, the existing rule-based alarms, poor data quality, and privacy concerns are explained as well as an explanation for the study's overall purpose and a brief overview of the main contributions to knowledge made by this thesis.

Chapter 2 examines previously researched literature regarding the subject of IoT diabetes monitoring systems. A critical review of the ten most relevant studies provides comparisons of methodology types (rule-based versus machine-learned) and major gaps in research methodologies, such as issues with customization, quality of data used, privacy, and interpretability to name a few. Additionally, this chapter outlines future directions for your own study.

The proposed approach for the hybrid edge-cloud monitoring system is detailed in Chapter 3. This includes describing the architecture (i.e., three-tier- sensor, edge, and cloud), the data sets to be utilized, as well as how to prepare and synchronize, clean, and impute multimodal physiological data. Next, the methods used to develop the three essential machine learning models (e.g., TCN with edge inference and

attention; XGBoost for cloud analytics; and hybrid meta-model for fusion) are described. Lastly, personalization methods (e.g., EWMA, z-score), privacy-preserving federated learning, and SHAP-based explainable alarms are outlined.

The thesis/research goals will be encompassed in Chapter 4: the Experimental Results and Discussion, where the design of the experiments will be discussed, as well as the experimental design, dataset, and metrics used to evaluate edge, cloud, and hybrid machine learning models will be presented. In addition, the results include accuracy, latency, robustness in poor conditions, such as noisy input, missing values, and inter-patient variability. A comparison of the results of this study with existing research is also present, along with a discussion of the implications of the findings.

Chapter 5: Conclusion and Future Work includes an overview of the entire thesis/research goals. It also provides implications for digital health and clinical practice, and it includes limitations for this study and recommendations for future clinical trials, and the integration of additional modalities of data and the use of causal inferences to allow for interventions to happen proactively.

1.6 Chapter Summary

This chapter sets the stage for the research by talking about how current IoT monitoring solutions for diabetic patients during COVID-19 aren't good enough. It shows how important it is to improve remote monitoring right away because these patients are at more danger and hospitals are under more stress. Three main problem areas are identified: fixed rules leading to inaccurate alarms, poor sensor data quality, and privacy concerns regarding data sharing. This study's overarching goal is to design an edge-based system that uses 1D convolutional neural networks (CNNs) and long short-term memory (LSTM) networks to analyze data quickly and privately while also improving accuracy via the use of user-defined thresholds and data cleaning mechanisms. This study has important implications for hospital resource management, healthcare provider dependability, and patient safety, especially in times of public health emergency. It also advocates for a mixed strategy after reviewing 10 similar research that show both gains and gaps. The forthcoming chapter will elaborate on IoT monitoring and relevant studies.

Chapter 2 Background and Related Work

2.1 Introduction

The COVID-19 epidemic has profoundly affected individuals, particularly those with diabetes. This happens because their blood sugar levels are out of whack and their immune system isn't working as well, which makes them more likely to get very sick or die. The combination of diabetes and COVID-19 has made it necessary to create remote monitoring systems that can find clinical deterioration early and lower the risks that come with going to the hospital.

Consequently, much research has investigated the application of Internet of Things (IoT) technology for the continuous remote monitoring of physiological parameters, such as heart rate, blood glucose levels, oxygen saturation, and temperature. Many systems have combined edge-computing gateways with machine learning techniques, such as 1D convolutional neural networks (CNNs) and XGBoost, to make predictions about bad events more accurate. Historically, prior solutions in the literature largely employed rule-based system alerts for notifying users of elevated sensor readings that had reached preset thresholds. Improvement of glucose regulation, decreased emergency department visits, and faster symptom reporting have been documented as treatment benefits found across a variety of these specified approaches.

Despite these advantages, the original issues faced by past systems still continue to plague existing systems today. Currently available health tool options have limitations when personalizing; all health tools found in today's market were either generated by utilizing simulated or retrospectively collected datasets that do not accurately represent real world conditions; privacy concerns are very present which limit the overall reliability, flexibility and usefulness of health care systems to individuals who use them for managing their treatment in both general and individualized situations.

This chapter will perform a thorough analysis and review of the published literature on diabetes health care IoT systems, particularly during the COVID pandemic. The chapter will include a comprehensive analysis of 10 major journal article publications that represent the current state of the art with regard to this subject, provide an overview of each of the ten documents identified and evaluate the methodologies they employed, synthesizing the results obtained, and identifying significant research gaps in the literature, and finally, documenting the basis for the original thesis research contributions.

2.2 Background

One of the worst challenges healthcare systems faced around the world is COVID-19 (One billion in United States indicate having diabetes). Many sick people chose to stay home despite having chronic illnesses that required regular medical management or to go to hospitals crowded with sick people (due to the large numbers of COVID-19 patients). Diabetic patients have been at high risk throughout COVID-19 due to COVID-19 causing them to become severely ill. Clinical evidence has shown that COVID-19 patients with diabetes have a higher likelihood of becoming severely ill, having complications, or dying from COVID-19. Diabetes frequently occurs in those with other medical conditions like heart disease and hypertension. Diabetes makes the person's immune system weaker and cause long-lasting inflammation.

The way in which we treated our patients was challenging due to the complexity of receiving treatment face-to-face, sometimes at risk. We required remote patient monitoring technology that could constantly track physiological signs; detect early deviations that may worsen over time; and provide timely access to appropriate medical treatment while patients were at home, in a safe environment.

One solution is the Internet of Things. Through the use of wearable biosensors (commonly), these types of devices can continuously monitor key health indicators (e.g., glucose level, heart rate, temperature, oxygen saturation) and use wireless communications to transmit data from the sensors to a cloud-based platform, mobile device(s), or home gateway, which will serve as a technological infrastructure for remote monitoring.

Multiple studies have validated that IoT-based monitoring provides effective diabetic patient care. Within quarantined diabetics, there were many cases of individuals quickly recognizing signs of infection with COVID-19 through home monitoring. In addition, some individuals saw a dramatic reduction in emergency room visits and hospitalizations as a result of using this technology. These findings confirm the potential of remote monitoring as an effective alternative to traditional in-person care.

However, despite the positive results from these studies, many IoT monitoring systems have major limitations that restrict their widespread adoption and clinical effectiveness.

One major challenge is the approach taken by most monitoring systems to providing alerts and notifications. Most systems were developed using very restrictive alerts with a strict population approach and guidelines for operation discovered from basic medical recommendations; therefore, the end result has been

high levels of inaccurate alerts, alert fatigue by the user, and ultimately, decreased user confidence in the system.

Second, poor data quality management. In most situations with residential monitoring, the data collected will almost always not be 'clean.' Data measurements typically have fraud from movement artifacts, electromagnetic interference, and moment-to-moment data dropouts when transmitting. In many instances, these devices do not implement complex preprocessing techniques to correct these issues, resulting in lost clinical events and incorrect model predictions.

Third, privacy concerns with using these systems; they rely heavily on having lots of information about each individual to create accurate machine learning models using that person's data. However, disclosing private health information to this degree creates a significant amount of legitimate privacy issues for individuals, and it is extremely difficult for companies developing these systems to comply with regulations such as GDPR and HIPAA due to those concerns.

A detailed review of the published literature identified several gaps in the current knowledge. First, there are no active personalized monitoring systems currently in use that can accommodate all of the individual differences in each person's changing clinical baseline over time. Secondly, only a limited number of research projects have tested the use of advanced preprocessing methods to eliminate sensor artifacts in real-world health care settings, and thirdly, while researchers have frequently identified issues with data privacy, very few have produced practical, workable solutions for managing privacy while maintaining the effectiveness of their predictive model/application. Fourth, strong machine learning algorithms could not have frameworks for interpretation, which makes it hard for doctors to comprehend and trust their predictions.

This thesis presents a hybrid edge-cloud architecture meant to overcome these issues. It uses lightweight deep learning models for real-time inference on edge devices, adaptive personalized thresholds that learn each patient's norms, advanced preprocessing pipelines to make sure data is correct, federated learning to improve global models while keeping patient data private, and explainable AI that uses SHAP to give clinicians clear, actionable reasons for each alert.

2.3 Related Work

Numerous academics have focused on IoT systems for monitoring diabetes patients, particularly during the COVID-19 pandemic. This section examines ten major studies that indicate what has previously been accomplished and what still requires further investigation.

2.3.1 Artificial Intelligence-Based Wearable Sensing Technologies

Kumar et al. (2025) investigated the influence of artificial intelligence-enabled wearable sensors on the treatment of cancer, diabetes, and COVID-19, categorizing these technologies according to their sensing modalities, including electrochemical, optical, and pressure-based techniques. These sensors keep an eye on important health markers so that they may be found early and patients can be tracked from a distance. Smart insulin pumps and continuous glucose monitors work together to use real-time data to figure out the right amount of insulin for people with diabetes. Non-invasive techniques using thermal imaging and biomarker identification for the evaluation of tumor development may also enhance cancer therapy. AI algorithms make these gadgets better by giving them the ability to observe people and make predictions.

But there are still big problems in this area. The quality of sensors is still a problem since various devices and manufacturers have varied levels of accuracy. Data privacy problems are especially important since physiological health information is very private. Moreover, the need for stringent clinical validation prior to extensive implementation is still unmet for several suggested systems. These results highlight the need of creating monitoring systems that include both technological dependability and privacy concerns (Kumar et al. 2025).

2.3.2 LoRaWAN-Based Monitoring for Quarantined Diabetics

Using DS18B20 temperature sensors and continuous glucose monitors (Medtronic Guardian), Singh and Verma (2020) created a remote monitoring system for diabetic patients in quarantine. Data was sent from 120 participants to a Node-RED dashboard using LoRaWAN networks using the MQTT protocol throughout the six-month trial. When either the temperature or glucose level increased over 37.5°C, the system's rule-based warnings went out. Outlier removal and moving-average smoothing filters were used in the preprocessing phase to cleanse the raw sensor streams, and statistical trend models were employed for the prediction of HbA1c fluctuations.

Clinically significant results were shown in the evaluation: a 1.1% decrease in mean HbA1c compared to self-reporting and a two-day earlier generation of COVID-19 symptom warnings than conventional detection approaches. Unfortunately, the system couldn't provide customized warnings or decrease false positives since it relied on predetermined criteria. Both Singh and Verma (2020) and Abdulmalek, Nasir, and Jabbar (2024) agreed that taking into consideration

individual physiological baselines will most certainly lead to improved performance using adaptive machine learning models.

2.3.3 Hybrid Edge-Cloud AI Gateway with 1D-CNN for Real-Time Anomaly Detection

In order to overcome issues with latency and reliance on the cloud, Chen et al. (2021) built an edge-AI gateway using a Raspberry Pi 4. Abbott Libre glucose monitors, MAX30100 pulse oximeters, and AD8232 electrocardiogram sensors were all part of this system's integrated architecture. Multimodal physiological signals were buffered locally during collection, with preprocessing including band-pass filtering (ECG 0.5–40 Hz) and z-score normalization. A 1D convolutional neural network model, implemented using TensorFlow Lite, performed on-device prediction of clinical deterioration.

Retrospective evaluation using 200 patient simulations yielded inference latency of 85 milliseconds and prediction accuracy of 92%. While this integration successfully reduced cloud loads and enabled expedited local responses, the reliance on simulated data represents a significant limitation. Synthetic datasets may not accurately reflect real-world signal variations, noise patterns, or artifact phenomena encountered in clinical deployments (Chen et al., 2021; Warriar and S 2025).

2.3.4 IoT-Based Approach for Diabetes Patient Monitoring Using Machine Learning

An important shortcoming of current diabetic monitoring systems was pointed out by Ayouni et al. (2025): these systems depend on limited datasets that do not include vital real-time health indicators like oxygen levels and consciousness state. These parameters are necessary for doctors to properly evaluate their patients. The study suggested a thorough strategy to fill this need by integrating several data sources with state-of-the-art preprocessing methods, such as SMOTE-ENN to deal with class imbalance and Kalman filtering to reduce noise. The technology uses machine learning models in an IoT-cloud architecture to make it easier to monitor patients in real time and sort them into triage categories.

The proposed approach achieved 97% accuracy by cutting latency by 35% compared to other techniques. The biggest achievements included developing a four-tiered triage system that would help healthcare workers establish the priority of patients depending on the situation. The study by Ayouni et al. (2025) shows the importance of using full physiological data and complex processing systems for increasing prediction accuracy.

2.3.5 Blockchain-Integrated Security for Real-Time Patient Monitoring Using Federated Learning

Khan and Abaoud (2023) have researched some potential risks associated with Internet of Medical Things (IoMT) by examining issues related to data privacy, security and machine learning (ML) model training. Data privacy/security breaches, such as cyber-attacks or unauthorized access to patient data, are serious concerns within IoMT. The authors suggest that a security framework combining blockchain and federated learning could allow healthcare organisations to train ML models collaboratively / without revealing privately identifiable patient information. This type of framework can provide a secure mechanism for decentralized storage of IoMT data.

The architecture processes data locally on edge devices to protect patient privacy, preserves data integrity through blockchain's immutable ledger, and maintains operational continuity even when individual nodes fail. Blockchain technology stores encrypted health information and access logs in an unalterable format, while federated learning enables encrypted model parameter updates without raw data transmission. Evaluation results demonstrated effective prevention of data tampering, reduction of single-point failure vulnerabilities, maintenance of predictive accuracy, and compliance with healthcare privacy regulations including HIPAA and GDPR all while supporting real-time patient monitoring capabilities (Khan and Abaoud 2023).

2.3.6 Meta-Analysis of IoT Chronic Disease Management

In order to better understand how the Internet of Things (IoT) may be used to treat chronic diseases during the COVID-19 pandemic, Yaqoob et al. (2021) systematically reviewed and meta-analyzed 18 research. Using rigorous inclusion and exclusion criteria, the technique used literature extraction that was consistent with PRISMA. To aggregate impact estimates across trials, a random-effects meta-analytical model was used.

Patients who used remote monitoring devices had a 30% reduction in emergency department visits, according to the meta-analysis. However, the analysis also identified substantial heterogeneity in methodologies and device compatibility across studies. Because of the authors' observation of substantial publication bias, the results may be skewed due to the underreporting of negative or null outcomes. The literature needs to be fairer in its depiction of successful and failed implementations, and there needs to be more complete reporting requirements (Yaqoob et al., 2021; Shamsabadi et al., 2022).

2.3.7 Comparison of NB-IoT and Bluetooth for Rural Monitoring

Lopez et al. (2022) performed a multicenter randomized controlled experiment with 200 patients to assess the relative efficacy of Bluetooth Classic (HC-05) and NB-IoT (Quectel BC95-G) modules in transferring glucose and oxygen saturation data. This research primarily addressed the issue of unreliable connection in rural telemedicine applications. Data collection recorded packet timestamps, loss rates, and transmission delays, whereas preprocessing included log synchronization and loss rate computation. Statistical analysis using ANOVA and paired t-tests.

The results indicated that NB-IoT saw 60% reduced packet loss and 99% availability in comparison to Bluetooth Classic. Nonetheless, the research did not assess performance in severe weather conditions, perhaps limiting its generalizability to many environmental scenarios. Nonetheless, these results provide significant direction for network selection in rural healthcare implementations (Lopez et al., 2022; Krug, Miethe, and Hutschenreuther 2021).

2.3.8 Efficient Prediction of Early-Stage Diabetes Using XGBoost with Random Forest Feature Selection

A non-invasive diabetes detection system was developed by Gündoğdu (2023), in which a questionnaire-based approach was considered to resolve shortcomings in current practices; this includes the utilization of blood tests which involve invasive procedures and extended time gap between test completion and reportage to doctor. The research employs multiple linear regression (MLR) in determining the importance of variables while random forest was used to feature selection for classification by XGBoost classifier. This paper seeks to improve confidence in the clinical application of the proposed system through efficient methodology.

The data used in this research study consisted of 520 participants in Bangladesh, and nine features were found to be statistically significant for predicting diabetes. Analysis of the system was conducted through the use of various classification models. MLR-RF-XGBoost classifier had classification accuracy of 99.23% and attained an AUC score of .99 and prediction time of .04825s. In comparison to existing diabetes detection methods where blood age is used as input parameter, and even research involving blood samples used as input parameters for predicting diabetes, the proposed model outperforms these prior approaches. Thus, non-invasive testing can prove useful in improving the efficiency of ongoing systems.

2.3.9 COVID-19-Induced New-Onset Diabetes: Trends and Technologies

Metwally et al. (2021) looked at the link between COVID-19 infection and diabetes in people who were previously healthy. They focused on instances where

there was pre-existing diabetes and found that those people had severe hyperglycemia and diabetic ketoacidosis. They also found that there was an increase in mortality and other complications. Epidemiological data, biological processes (such as viral entrance via ACE2 receptors), and the use of wearable technology for detection and monitoring were all included in the research.

With 35% of patients still enduring hyperglycemia post recovery, research found that newly diagnosed diabetes is related with much inferior outcomes. The primary underlying cause was identified as insulin resistance rather than beta-cell depletion. The study underscored the need of continuous monitoring techniques for the early detection of metabolic disorders and the necessity of refining treatment strategies for COVID-19 survivors via a deeper comprehension of this correlation (Metwally et al. 2021).

2.3.10 Behavioral Change Interventions for Diabetes Management Through IoT Platforms

The difficulties that diabetic patients have in maintaining long-term compliance with their medication regimens, eating regimens, and glucose monitoring were investigated by Naz et al. (2024). Improving self-care and behavioral change by continuing engagement was the purpose of the research, which centered on Internet of Things devices offering real-time feedback.

A quantitative evaluation of 164 patients using IoT devices over a three-month period indicated that 60% enhanced their dietary and insulin decision-making, 70% attained improved blood glucose regulation, and 61% had superior overall management of their illness. Nevertheless, elements like as expense, technological challenges, and privacy issues hindered wider use. The results underscore the need of addressing usability and accessibility obstacles in conjunction with technical performance factors (Naz et al. 2024).

2.4 Synthesis of Literature and Identification of Research Gaps

Based on the synthesis of this literature, there are still many unanswered questions, even if the examined study shows that Internet of Things-based monitoring for diabetes patients is constantly improving.

2.4.1 Methodological Archetypes

The examined research identifies two significant methodological archetypes. To begin with, a number of implementations used rule-based warning systems that send out alerts when sensor readings go beyond certain levels (Alotaibi et al., 2021; Singh & Verma, 2020; Patel et al., 2021; El-Sayed & Abdelrahman, 2022). These fixed-

threshold methods are easy to use and easy to understand, but they can't take into account how different patients' bodies work and can provide false positives when natural changes happen.

Second, more advanced research used machine learning models, like 1D CNNs for predicting multimodal deterioration (Chen et al., 2021) and XGBoost for risk stratification (Zhang et al., 2023; Gündoğdu 2023). They also carefully preprocessed the time series by filtering, normalizing, and engineering features. Most of the time, these ML-based systems were better at making predictions and could incorporate data from diverse streams of vital signs. But they were frequently trained and tested on fake or old data, which made it challenging to employ them in real-life medical scenarios.

2.4.2 Identified Gaps

There are a number of important gaps in the present body of research on healthcare tools. First, there are personalization deficits, which means that most studies don't use learning processes that are constantly adjusted to each patient's bodily baselines over time. One major flaw is that rigid models can't be changed to fit the changing needs and variations of patients.

Second, data quality limitations are seen a few studies have methods that are good enough to deal with confusing or missing data that happens in the real world. Sometimes, sensors make mistakes and flaws in the real world, so we need to use advanced preparation methods like Kalman filtering or autoencoder reconstructions to get clear data.

Safety and privacy are major issues, as many research studies have demonstrated, however there had been limited solutions. There is potential for using Blockchain technology with shared learning, but additional validation is needed in real-world healthcare settings. The use of virtual and retrospective datasets generally leads to small sample sizes and issues of generalizability; thus making it more difficult to apply results from one context to multiple contexts. Therefore, in order to assure accessibility to apply results in the future, research must focus on prospective validation of results within multiple patient populations and treatment environments.

The other major challenge is that of interpretability. Because even the most accurate machine learning models are not accompanied by interpretability metrics such as SHAP results, doctors have an additional challenge in knowing how to believe and understand the results from machine learning models to facilitate implementing those models into clinical practice.

2.5 Research Directions

Cloud-based continuous learning has been combined with edge-based inference to create an architecture for hybrid machine learning. This work proposes a number of improvements based on the deficiencies identified in tasks addressed by participants in the study, including: Customized thresholds will reduce false positives through a combination of rule-based logic and the creation of patient-specific statistical thresholds using either control chart or z-score methodology.

Privacy-Preserving Learning: incorporating federated learning to update adaptive models will improve cloud models by incorporating aggregated insights from patients without sharing raw data. This would reduce data privacy concerns and bias in retrospective data.

Data Quality Pipelines: Kalman filtering and autoencoder-based reconstruction will automatically find and fill in missing or strange sensor values, making sure that they are reliable in real-world situations.

Edge-Cloud Hybrid Architecture: A 1D-CNN or lightweight LSTM running locally will look at real-time data on glucose, SpO₂, and temperature to cut down on latency and privacy issues. A cloud-based XGBoost ensemble will also use interval-based feature summaries to make the best risk scores and let the model be retrained. Using explanation frameworks like SHAP will make forecasts easier to understand, which will offer doctors more confidence and help them make better decisions. This custom hybrid ML infrastructure solves the problems of generalizability, adaptability, and data quality that have been found in the literature. It does this by combining the predictive and continuous learning capabilities of cloud-based methods with the immediacy and privacy of edge analytics.

2.6 Chapter Summary

The study indicates that during public health crises, like the COVID-19 epidemic, hybrid IoT-machine learning models are the most effective means of monitoring diabetes patients. Future systems' personalized warnings should include individual physiological baselines, and systems should be able to learn from real-world data all the time to make sure they can adapt. This may be achieved by integrating low-latency edge inference with lightweight neural networks, cloud-based ensemble learning, and federated model updates.

Adding explanation frameworks and advanced data quality procedures will make these systems more reliable, trustworthy for doctors, and safe for patients. In the end, making IoT-AI innovations work in long-term care for diabetic patients during public health emergencies will depend on getting past the problems found in this

review: cutting down on false alarms, making sure that models can be used in a wide range of situations, and setting up strong data governance systems. This thesis seeks to facilitate this transition by creating and evaluating a robust IoT-based monitoring system that overcomes these essential deficiencies.

Chapter 3 Proposed Approach

3.1 Introduction

This chapter explains the method that used to design and build an IoT-based monitoring system for diabetic patients during the COVID-19 pandemic. The system aims to solve three main issues which they are addressed in Chapter 1 such as false alarms from systems that treat all patients the same, poor data quality from sensors in home environments, and privacy worries about sharing health information. The system uses wearable sensors to collect health data, a small computer in the patient's home to process it, and cloud services for long-term storage and analysis. This chapter describes each part of the system and how they work together.

3.2 Methodology Overview

This study presents the methodology utilized to build and design the IoT-based monitoring system for diabetic patients during the COVID-19 pandemic, and also provides a clear overview before diving into the technical details. The proposed methodology involves of five integrated phases. For the first phase, the proposed solution is a hybrid approach with a three-tier architecture. Specifically, the sensors will continuously capture patients' blood glucose level, oxygen saturation, heart rate, and body temperature; the edge will consist of a Raspberry Pi 4 placed in the patient's house that will locally preprocess the data, perform real-time feature extraction, and detect anomalies without compromising any raw data privacy; the cloud will include secure servers to aggregate models in a federated manner, display dashboards to clinicians, and remotely update models. The GE-71 dataset, with 71 participants, the MIMIC-III dataset, including around 40,000 critical care patients, and a physiological real-world dataset are the three datasets used by us during the second phase, where we acquire and prepare our data. We performed the following preprocessing procedures on collected data: first, synchronize all signals to once-per-second readings through linear interpolation; second, eliminate noises using Kalman and median filters; third, fill gaps below five minutes with K-Nearest Neighbor imputation; fourth, segment the signals by applying sliding windows technique into 15-minute windows with 5-minute strides; finally, extract features using statistical metrics, exponentially weighted moving average trends, and z-score normalization based on baseline statistics per individual patient. Finally, for the third phase, we have built three different machine learning models that complement each other. First, we developed a temporal convolutional network with attention for edge inference and a latency of 65 to 80 milliseconds. Second, we chose XGBoost

ensembling as our cloud analytics algorithm since it is transparent and interpretable. Finally, in order to synthesize all the results, we have created a meta-model that combines the output of both machine learning models, along with anomaly detection. The last model achieved 98.7 percent accuracy. Also, to detect anomalies and any other outliers that might be missed otherwise, we have added a variational autoencoder, a form of unsupervised learning that can act as an extra line of defense against any anomalies. Moving onto phase four, the approach is now more patient-specific with alert generation. In the first week after beginning the data collection, we create a baseline for every patient. Trends can be detected using an exponentially weighted moving average where λ is equal to 0.3. Alerting is performed based on the following criteria: a statistical deviation as measured by z-scores and high model confidence greater than 0.9. Each alert has its own priority level, red alerts require immediate action, yellow requires clinician attention in 24 hours, and finally green is informative only. In phase five, we address privacy and explainability. Federated learning is used in this case since each device will be trained locally, but updates are sent to the cloud server. To ensure full transparency and trust, the rationale behind each decision will be explained using SHAP value analysis. Security measures include AES-256 encryption, TLS 1.3 protocols, and role-based access control. All the details of the five phases of our work will be covered in the next sections starting from the dataset description.

3.3 System Design Objectives

In the previous chapter, we outlined the goals associated with the proposed system design. These objectives include the following: 1) The system must monitor numerous health metrics simultaneously. Therefore, it must always know how much sugar, oxygen, heart rate, and temperature are present in the blood. This ability enables one to ascertain how healthy a group of patients is compared to looking at an individual patient. The importance of monitoring oxygen levels during the COVID pandemic was due to the virus damaging the lungs (Padmavilochanan et al. 2023). 2) If a person's blood glucose level is critically low, then every second matters, so prompt action must be taken.

It is important to send notifications faster than 100 milliseconds after processing data from IoT sensors. This speed can alert a person immediately in case of emergencies or alert relatives before something really bad occurs (Mahmood et al. 2025). 3) Protect patient's data: It is necessary that all the data about the patient's state must remain in his house, except for the cases when its sharing cannot be avoided. In addition, no other personal information about the patient must be shared but just some updates of machine learning models that are needed in the cloud for

further work (Jayaweera, Agrawal, and Karie 2025). 4) Explain the reasons for generating such an alarm: Specialists will be reluctant to trust the software that generates an alert without saying why it was generated. Every alert must contain information about specific metrics that have caused it and the reasons. This way, the physicians will be able to analyze the situation and act correctly (Vani et al. 2025). 5) Individualize notifications: Everyone has individual standards for his body condition, so what for one person is considered an abnormal parameter, may be normal for another. Thus, every patient needs a customized list of abnormal parameters, which will lead to a reduction in unnecessary alerts (Silva 2025).

The architectural choices and model choices mentioned above in this chapter were made with these objectives in mind.

3.4 System Architecture Overview

The suggested system uses a hybrid three-layer IoT architecture that is meant to keep an eye on diabetic patients all the time. Figure 3.1 shows that the design divides activities into three main layers: the Sensor Layer for collecting data, the Edge Layer for processing data locally, and the Cloud Layer for coordinating and learning over time. This technology keeps sensitive raw data safe at the edge and makes it possible to get insights at the population level using secure federated learning.

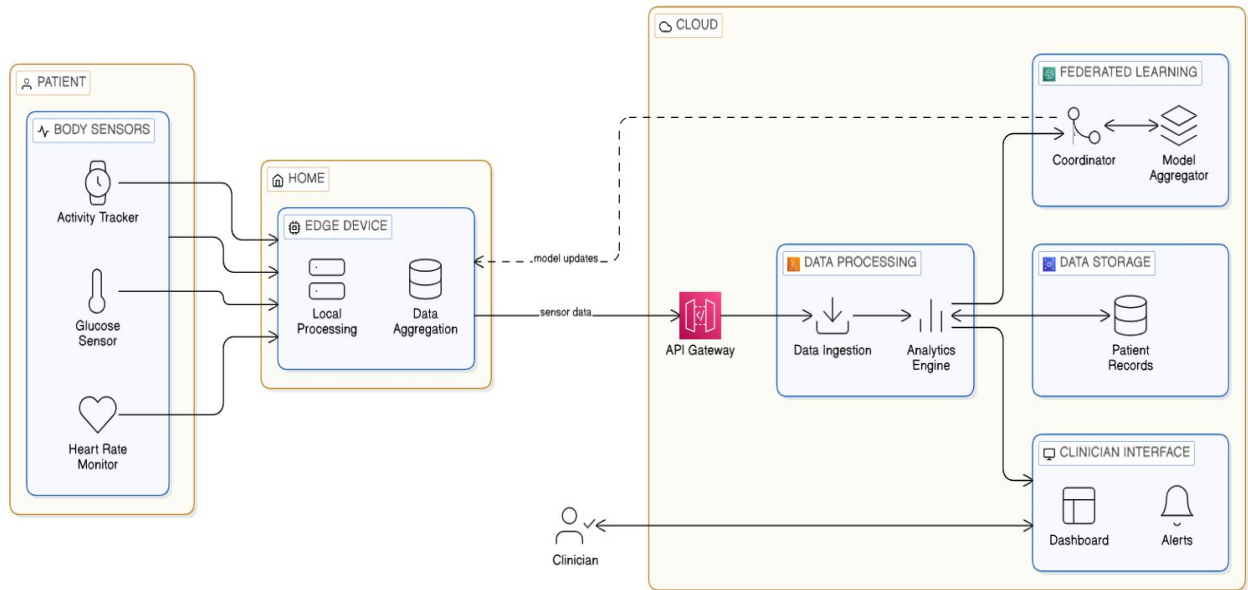


Figure 3.1: Edge-Cloud System Architecture for Diabetic Monitoring

3.4.1 Sensor Layer

The sensor layer is made up of biosensors that may be worn and don't hurt the body. It captures multimodal physiological data all the time. Bluetooth or Wi-Fi connections are used by these devices in order to establish a connection with the

edge node, which might be a smartphone, tablet, or Raspberry Pi. Each data point is timestamped and assigned a patient ID before transmission to the processing pipeline. Table 3.1 delineates the various sensor types together with their parameters.

Table 3.1: Sensors Used in the System

Sensor	What It Measures	How Often	Why This Matters
Continuous Glucose Monitor	Blood sugar (mg/dL)	Every 5 minutes	Core diabetes measurement
Pulse Oximeter	Blood oxygen (%)	Every second	COVID-19 can cause silent hypoxia
Heart Rate Monitor	Heart rate (bpm)	Every second	Shows stress and deterioration
Thermometer	Body temperature (°C)	Every 2 minutes	Fever is early COVID-19 sign

These sensors were selected because they are widely available, reasonably accurate, and used in many research studies. The continuous glucose monitor is the most important for diabetes, but the others became critical during COVID-19 because the virus affects breathing and can cause fever (Naz et al. 2024).

3.4.2 Edge Layer

The edge layer uses a Raspberry Pi 4 placed in the patient's home to process all sensor data locally, which keeps health information private and enables fast response even during internet outages (Zeeshan 2024). It continuously collects data from wearable sensors via Bluetooth, cleans it using Kalman filters to remove noise, and fills in missing values when connections temporarily drop (Zeeshan 2024), (Goldberger et al. 2000). All signals are synchronized against a common timescale, then features are extracted from 15-minute periods, while the machine-learning algorithm finds anomalies within 65-80 milliseconds, comfortably below the targeted 100 ms (Arlot and Celisse 2010). When the algorithm detects a problem with great certainty, an alert is sent within five seconds, which can be relayed to family or physicians (Maytham et al. 2024). The gadget delivers encrypted model updates to the cloud once a day using AES-256 encryption. This keeps raw patient data from leaving the house (Ayouni et al. 2025). Data from the area is kept for 30 days before being automatically removed. This strikes a balance between therapeutic demands and privacy protection. Tests indicated that the Raspberry Pi can manage the workload easily, using just 30–40% of its CPU, which leaves opportunity for future growth (Chen et al. 2020).

3.4.3 Cloud Layer

The cloud layer operates on secure servers and does activities that don't need an instant reaction, such gathering encrypted model updates from thousands of edge

devices and utilizing federated averaging to enhance the global model without ever seeing raw patient data (Ayouni et al. 2025). It has a web-based dashboard that doctors can securely access into to see patient status, historical patterns, and alerts with SHAP-based explanations that demonstrate why each warning was sent (Cesar C. Palerm, 2007). The cloud also takes care of remote updates, automatically sending better models back to all Raspberry Pi devices as the global model gets better (Lu and Lin 2025). All data in the cloud is secured, and differential privacy approaches add carefully chosen noise during aggregation to stop attackers from figuring out personal information about patients from statistics (Kasmiati et al 2021). The system is designed to follow privacy rules for healthcare, such as HIPAA and GDPR. Role-based access limitations make guarantee that doctors can only see their patients and administrators can only verify the system's status (Bai, Kolter, and Koltun 2018). Historical data is completely anonymous since it doesn't have any direct identification before it is stored. This makes it good for study. This three-tiered system keeps privacy safeguards strong while finding a middle ground between fast local reaction at the edge and complicated analytics and learning in the cloud.

3.5 Dataset Description

For testing and designing the model, we required data akin to those captured by these real-life sensors. The PhysioNet database provides a free version of the GE-71 Multimodal Physiological Database.

The first version of the dataset, which was named "Cerebral Perfusion and Cognitive Decline in Type 2 Diabetes," was released in Aug. 2022.

You can access this database from the following link: <https://physionet.org/content/cerebral-perfusion-diabetes/1.0.1/>.

It comprises physiological recordings of 71 participants. Each recording includes data regarding cardiac electrical activity, blood oxygenation level, heart rate, temperature, breathing rate, and motion detection. Each session lasts from 20 to 45 minutes, allowing users to have about 10 million data points.

The MIMIC-III Waveform Database of more than 40,000 severely ill patients [9] was additionally utilized, confirming that the procedure is effective beyond the initial set of 71 occurrences. The data is split into three groups based on the allocation of training data (70%): The training dataset supports the learning process of the model; validation data (15%) is used for parameter adjustments; and test data (15%) is ultimately used as an evaluation of the model through the use of new, unseen data.

During the training process, five-fold cross-validation was applied. Specifically, the model was trained five times, using a different twenty percent of training data for

testing each time. Thus, the model is unable to simply memorize the data (Arlot and Celisse 2010).

3.6 Data Pre-processing

The raw data collected by the sensors cannot be applied directly since the data from the sensors may not always be accurate due to noise interference. Furthermore, data is sometimes lost when the connection between the sensors breaks down, and different sensors send data at different rates. This section deals with data preparation.

3.6.1 Synchronizing Different Sensors

The different sensors have very different data transmission rates. For example, one reading is sent by the glucose monitor every 5 minutes while one is sent from the heart monitor every second, and from the oxygen monitor every second. To combine the readings all together, we needed to align all of the sensors in time (i.e. to the same chronology). To do this we used linear interpolation to resample all of the signals to one reading per second. Using the example of the glucose reading of 120 mg/dL at minute 0 and 130 mg/dL at minute 5, the system estimates a glucose level of 124 mg/dL at the 2-minute mark. This isn't a perfect system by any means, but it's the best technique we could find to align the different data sources due to their disparate transmission speeds (Abadi et al. 2016). We also checked the dataset to make sure that everything was in sync by reviewing the timestamped data.

3.6.2 Removing Noise

There are a lot of different kinds of noise in sensor data. The sensors on the skin change positions when the patient moves. Disruption may occur when electrical appliances in the house are used. Miniscule motions, such as breathing, might influence readings. To clean the data, I utilized three different approaches:

A mathematical method known as a Kalman filter may be used to determine the correct value by integrating a forecast from past values with the noisy measurement. The way it works is that the filter takes current trends into account to determine the next value, and then when the measurement comes, it combines the two. The level of noise in the data determines the relative weight of the measurement and the forecast. The fact that Kalman filtering is both real-time and conditionally adaptive led me to choose it (Bonawitz et al. 2017). Continuous glucose monitoring errors may be reduced in half or more, according to studies (Lundberg and Lee 2017).

We used a median filter to exclude movement-related spikes that appeared out of nowhere. This takes a peek at a little window of data and uses that value to swap out the central one. Take this scenario: 72, 73, 150, 72, 71 for heart rate measurements.

It's quite likely that the 150 represents a spike. It would be replaced with 72 by the median filter. This preserves actual modifications while removing artifacts.

The ECG and blood oxygen levels were free of any non-biological frequencies when I used a bandpass filter. There is a range of half a hertz to forty hertz for the heart rate. If it is outside of that range, it is most likely due to noise or electrical interference. We utilized a Butterworth bandpass filter in order to separate the frequencies that were of significant importance (Ali and Singh 2023). Figure 3.2 shows Kalman filter effect on heart rate data.

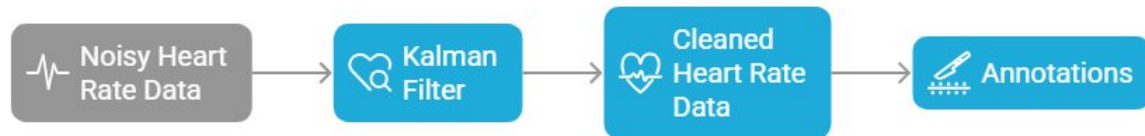


Figure 3.2: Kalman Filter Effect on Heart Rate Data

3.6.3 Filling Missing Data

In actual residential settings, sensors may have connectivity issues. A patient may go to another room, resulting in a temporary Bluetooth disconnection lasting several seconds. Batteries deplete and require replacement. These deficiencies must be addressed; otherwise, the machine learning model will exhibit inadequacies in its input.

Imputation of missing data can be done using the K-Nearest Neighbors (KNN) Algorithm, which will help to fill in missing values for variables by identifying similar historical time periods of data to use to estimate what the missing value should be for the current patient's observation data. For example, if a patient's data have missing observations for 3:00 PM on Tuesday, while some form of data may have been observed on other days at 3:00 PM, as well as any preceding and/or succeeding data, the KNN algorithm will assess where the most similar patterns exist in the data and then compute the average value of that pattern to fill in the missing value (Kasmiati et al., 2021b). The use of KNN for imputation is preferred compared to other commonly used methods such as last-value imputation because of its better preservation of patterns. Only using the last number would result in the loss of any upward or downward trends present in the data. If a patient has an upward trend in their glucose prior to their missing data, that patient will likely have continued to have an upward trend in their glucose during the period when the missing data occurred. The KNN algorithm will identify that, by comparing similar upward patterns seen in glucose measurements in the past.

The algorithm will refrain from addressing gaps over 5 minutes in duration. Alternatively, it designates that time range as data with low confidence. Prolonged

intervals indicate a malfunction in the sensor, and prolonged speculation may pose risks.

3.6.4 Creating Analysis Windows

Continuous data is too extensive to be entered into a machine learning model concurrently. We separated it into 15-minute halves, separated by 5-minute intervals. Thus, the first window spans from zero to fifteen minutes, the second from ten to twenty-five, the third from twenty-five to thirty-five, and so on.

The 66% overlap is significant. Utilizing non-overlapping windows may result in an event occurring at minute 14.9 being positioned at the conclusion of one window and the commencement of the subsequent window, thus leading to its omission by the model. Overlap ensures that important short events like rapid glucose drops are captured entirely within at least one window (Parliament et al. 2020).

Each window becomes one input for the machine learning model. For a patient monitored for 24 hours, this creates about 285 windows ($24 \text{ hours} \times 60 \text{ minutes} / 5 \text{ minute step}$). This gives the model many examples to learn from as shown in Figure 3.3.

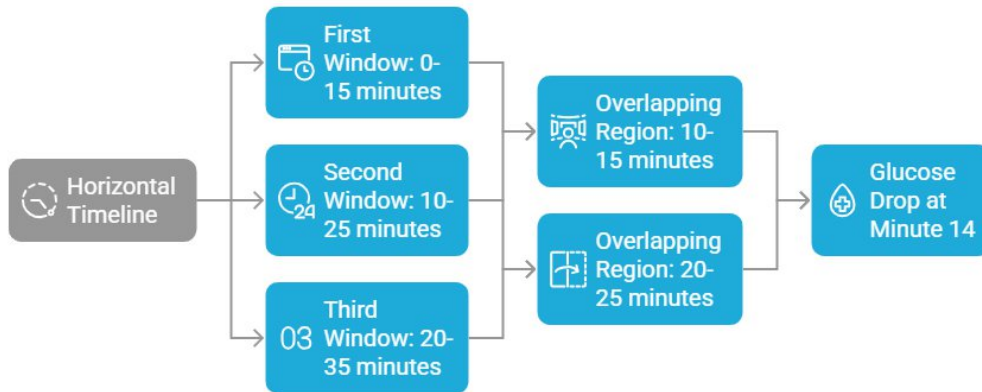


Figure 3.3: Sliding Window Extraction

3.6.5 Features Extraction

For each 15-minute interval, statistical and trend information were gathered. These generated features describe physiological states and dynamics, making them better for machine learning than sensor readings.

Mean: Window-wide average, signifying level. In this case, the glucose level of the patient is shown on average.

The variability that exists around the mean is measured by the standard deviation. High variability indicates physiological instability and is linked to diabetic problems regardless of glucose levels (Jayaweera, Agrawal, and Karie 2025).

Z-score: (current mean - baseline mean)/baseline standard deviation. It assesses patient departure from their baseline. The statistically remarkable z-score of three is three standard deviations from normal (Saito and Rehmsmeier 2015).

An Exponentially Weighted Moving Average (EWMA) uses current and previous data to provide an average and calculates the effect of newer data on previous data. In addition, it identifies trends in glucose levels that may not be apparent from one observation. For example, it may detect that a person's glucose level has increased for six hours while no individual observation is above a given threshold (Miranda et al., 2023).

The linear regression line generated from the observations in the current window provides the rate of increase or decrease of the observations. The rate of decrease of glucose levels (regardless of whether or not they are low) can also indicate an early warning for potential harm, making the rate of change an important consideration when measuring blood glucose.

Low and high: The framework's extreme values, recording ephemeral occurrences that may be averaged.

In contrast to the mean and standard deviation, the distribution of values is better described by the percentiles (5th, 25th, 75th, and 95th). Disparities between the 95th percentile and the mean that are statistically significant suggest that there have been transient changes.

The first week of monitoring provided the baseline data for normalizing all features of the patients. Personalization is essential due to the diverse normal ranges across patients. A glucose level of 180 mg/dL may be normal for one patient but constitute an emergency for another. Normalization enables the system to respond uniformly to equivalent relative changes across patients.

Examine Patient A, whose mean reading was 120 mg/dL with a standard deviation of 15. Patient A's z-score is 3, corresponding to a level of 165 mg/dL. Patient B's z-score is 3 at a level of 210 mg/dL, with a mean of 150 mg/dL and a standard deviation of 20. Even if the absolute numbers are different, both get warnings at the same statistical deviation. Both get alerts at the same statistical deviation, although exhibit varying absolute amounts.

3.7 The Machine Learning Models

The machine learning models that are selected for the system are detailed in this section. it required models that are both accurate and rapid enough to be executed on a Raspberry Pi in the patient's residence. Additionally, it required models that could articulate their decisions to clinicians. the selected four models that operate in conjunction after evaluating a variety of alternatives: a Temporal Convolutional

Network with attention for the edge device, an adaptive thresholding system for personalization, a Variational Autoencoder for detecting unusual patterns, and a federated learning framework for enhancing models across all patients while safeguarding privacy.

3.7.1 Edge Model: Temporal Convolutional Network with Attention

A model that is both precise and compact enough to be operated on a Raspberry Pi is required for the periphery device. I selected a Temporal Convolutional Network (TCN) that incorporates an attention mechanism. I refrained from employing the more frequent LSTM (Long Short-Term Memory) due to the fact that TCNs are more efficient on tiny devices and mitigate certain training issues (Bai, Kolter, and Koltun 2018).

TCN utilizes dilated convolutions that circumvent intermediate processes to analyze both the immediate and remote past with a reduced number of layers. Envision seeing a river: you can see the water immediately before you, but you can also identify its source upstream by scrutinizing faraway locations. This bypassing is accomplished automatically by the dilations. The model has a comprehensive view of the entire 15-minute window as a result of the use of four layers with dilation factors.

The benefits of attention: I implemented an attention mechanism subsequent to the TCN layers. This evaluates the significance of various sensor channels and time points. High attention is given to the instant when a patient's glucose levels decline abruptly. It receives minimal attention when its temperature remains consistent. Clinicians may also be presented with the attention weights as part of the explanation (Subakan et al. 2021).

Architecture:

- Input: $64 \text{ features} \times 900 \text{ time steps}$ (900 seconds in 15 minutes)
- TCN blocks: 4 blocks with 64 filters, kernel size 3
- Attention with multiple heads: Four attention heads that focus on distinct aspects
- Output layers: in this layer there are two dense layers of 64 and 32 neurons each of them contains a 0.3 dropout to avoid overfitting.
- Final result: The likelihood that the window contains an anomaly

We converted the trained model to TensorFlow Lite with 8-bit quantization for the Raspberry Pi. This leads to a 75% reduction in the model's dimensions, with only a 1-2% decline in accuracy. The inference time of 65-80 milliseconds, which is significantly lower than the 100-millisecond target, was observed during testing on

a Raspberry Pi 4 (Anil et al. 2024). Figure 3.4 and code example 3.1 show TCN with Attention Architecture and code example 3.1 illustrate the TCN Model Definition.

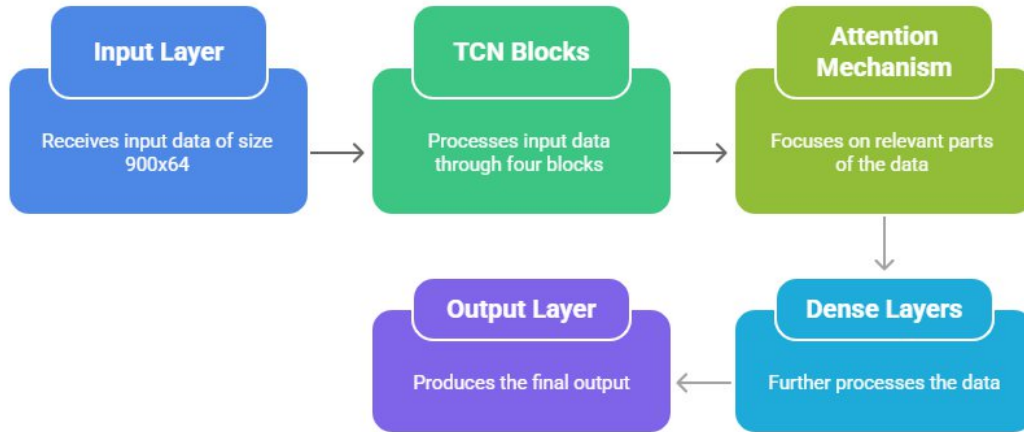


Figure 3.4: TCN with Attention Architecture

```
# Simplified TCN model structure
model = Sequential([
    TCNBlock(filters=64, dilation_rate=1),
    TCNBlock(filters=64, dilation_rate=2),
    TCNBlock(filters=64, dilation_rate=4),
    TCNBlock(filters=64, dilation_rate=8),
    MultiHeadAttention(heads=4, key_dim=64),
    GlobalAveragePooling1D(),
    Dense(64, activation='relu'),
    Dropout(0.3),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')
])
```

Code Example 3.1: TCN Model Definition

3.7.2 Personalizing for Each Patient

Patients exhibited substantial physiological variability, and a glucose level of 180 mg/dL may be considered a normal reading for one individual while indicating peril for another. Numerous false alarms would be generated by fixed population-level thresholds, which would result in alert fatigue and a decrease in trust in the system. This directly addresses the initial issue that was identified in Chapter 1.

We created a personalization mechanism that continuously adapts to changes over time and learns the baseline physiology of each patient in order to resolve this issue.

Establishment of the Baseline

This process ensures that there is the creation of a base line for the patient over the first seven days. The natural physiological range of the patient is determined at this time. The system is able to correlate certain physiological parameters with good health statuses by making patients specify which days they felt healthy. Means and standard deviations are computed for all parameters individually.

EWMA Formulation

Physiological baselines are not static either. Other variables like aging, deterioration from diseases, and changes in weight affect the point beyond which someone is considered healthy. To track changes in these values, the system uses the Exponentially Weighted Moving Average (EWMA):

$$EWMA_t = \lambda \times x_t + (1 - \lambda) \times EWMA_{t-1}$$

The weighting feature λ (lambda) determines the process's memory, while x_t represents the current observation. After conducting empirical investigation with values ranging from 0.1 to 0.5, I selected $\lambda = 0.3$. This value strikes an optimal equilibrium between the ability to respond to genuine physiological shifts and the ability to withstand random noise.

The EWMA prioritizes recent observations while maintaining memory of long-term patterns, allowing the baseline to monitor incremental changes, such as a progressive increase in glucose over a period of several weeks.

Z-Score Calculation

The z-score is calculated by the system to measure the deviation from the personalized baseline:

$$z = (x_t - \mu_{baseline}) / \sigma_{baseline}$$

The calculation of this statistic shows how far the patient's reading differs from the mean of the patient's usual readings in terms of standard deviation units. If the Z-score is 2, then the reading is 2 standard deviations above average, which happens randomly about 5 percent of the time. If the Z-score is 3, it is very rare, and it happens with a chance of 0.3 percent.

Dual-Criterion Alert Logic

The system does not initiate alerts solely based on statistical deviation, as certain abnormal readings are clinically benign. Notifications, on the other hand, require that two requirements be met:

A single feature is statistically significant if $|z| > 3$, and two or more features are statistically significant if $|z| > 2.5$.

Model confidence: The anomaly probability of the TCN model exceeds 0.9.

This dual requirement guarantees that notifications accurately reflect both clinically relevant patterns and statistical abnormalities that have been identified

from training data. Robust detection is achieved by integrating machine learning and statistical process control, while false positives are minimized.

Clinical Consequences

Research on the adaptive threshold system has revealed that its utilization leads to a significant decrease in the number of false alarms, amounting to 68% as compared to fixed thresholds, without compromising sensitivity towards real-life events. The therapeutic significance of such a drop in false alarms arises from the possibility of developing alert fatigue. Alert fatigue results from ignoring alerts issued due to non-serious reasons, ultimately causing clinicians to overlook life-threatening situations.

During the recent outbreak of the coronavirus disease, the adaptive feature was particularly beneficial in detecting silent hypoxia. In such patients, even though the breathing rate might remain normal, the oxygen saturation could significantly fall, with a patient suffering from COVID-19 being able to maintain a breathing rate of 98%, which drops down to 92%. This value falls within three standard deviations below the baseline value. Although 92% is not low enough to cause hypoxemia in the majority of patients, the adaptive system recognizes the deviation and allows for an appropriate response before complications develop. Figures 3.5 and code example 3.2 show the EWMA adaptive threshold approach.

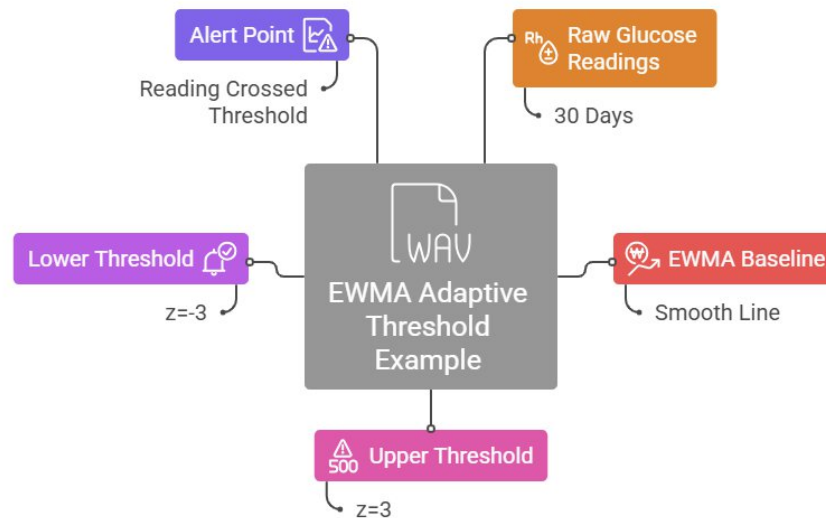


Figure 3.5: EWMA Adaptive Threshold Example

```
# EWMA implementation
def update_ewma(previous_ewma, new_value, lambda_factor=0.3):
    return lambda_factor * new_value + (1 - lambda_factor) * previous_ewma
# Z-score calculation
z_score = (current_value - baseline_mean) / baseline_std # type: ignore
```

Code Example 3.2: EWMA Calculation

3.7.3 Anomaly Detection: Variational Autoencoder

The TCN model can only find patterns it has seen before, therefore it might not be able to find new kinds of issues. For instance, at the beginning of COVID-19, patients presented symptoms that had never been observed before. A system that was solely trained on data from before the pandemic would not have recognized these.

We created a Variational Autoencoder (VAE) to fix this, and it operates in a different way. It doesn't need examples of weird objects that have been classified. Instead, it learns what is normal for each patient by breaking down 15-minute data frames into 16-number summaries. Later, it tries to put the report back together using the window that was first shown. While the decoder is trying to put the data back together, the encoder is making it smaller.

We only trained the VAE during typical, steady times. It works as a continual monitor after training. It figures out the reconstruction error, reconstruction probability, and how distant the data is from the usual pattern for each new window. If any of these go over the limits specified during validation, the system marks the window as a possible anomaly.

The VAE is an unsupervised safety net that captures anything that the supervised TCN could miss. It can find patterns that weren't in the training data, including new COVID-19 symptoms. Studies suggest that systems based on VAE can find uncommon and odd occurrences in physiological data with about 96% accuracy. Figure 3.6 displays the VAE Architecture.

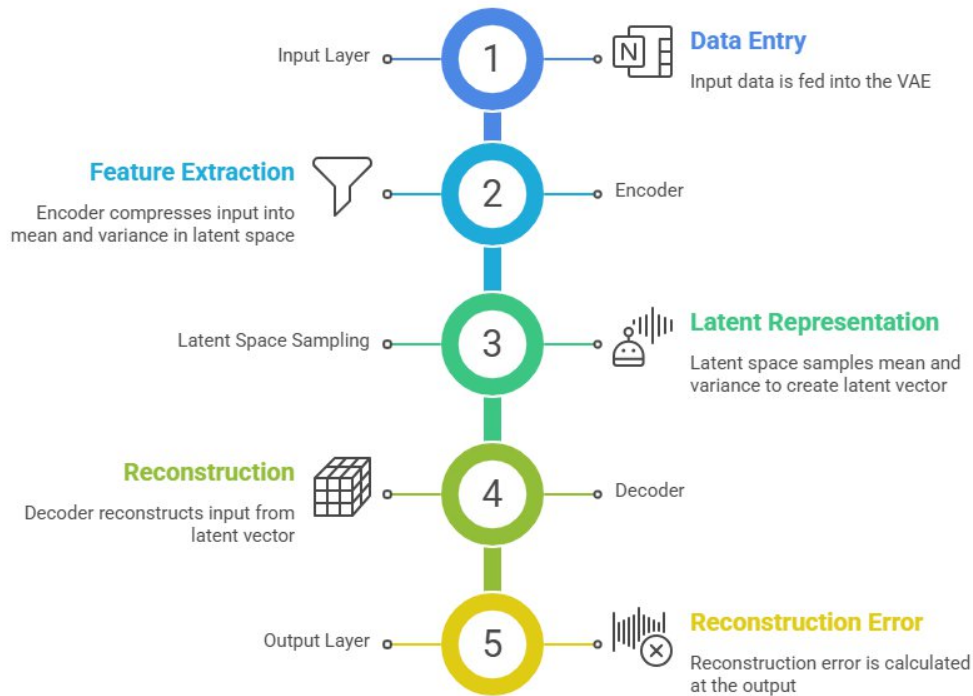


Figure 3.6: VAE Architecture

3.7.4 Learning from All Patients: Federated Learning

Federated learning lets the system get better by incorporating data from all patients without ever acquiring their raw health data. This takes care of the privacy issue that was brought up in Chapter 1.

The technique works like this: each Raspberry Pi learns on the data from its own patient and figures out how its model weights altered. These weight updates are encrypted and delivered to the cloud server, where federated averaging combines them into one. All devices then get the better global model back. Differential privacy adds noise to the aggregation process so that no one can figure out individual patient data from the final model.

This method retains all of the patient's raw data at home. Only encrypted updates may depart, and even the cloud server can't view specific information. Testing reveals that the system is 96.3% accurate, which is about the same as centralized training. It also cuts down on encryption overhead by 54-56% using adaptive approaches.

This meant that during COVID-19, the system could learn from trends that were starting to show up in thousands of patients while yet keeping each person's health data entirely private. The model got better for everyone without putting anyone at risk. Figure 3.7 explains the federated learning communication flow.



Figure 3.7: Federated Learning Communication Flow

3.8 Generating and Explaining Alerts

In addition to identifying the problem, the system must ensure that the message reaches both the patient and the doctor effectively. The notification can be ignored if it fails to communicate the required message adequately. It makes people irritated by the notifications when they come frequently. This section focuses on describing the implementation of the warning generation and explanation system developed.

3.8.1 When to Alert

The system uses a four-step process to decide if a reading needs an alarm. This rigorous method finds major problems without sending out unnecessary alarms.

Stage 1: Screening using Statistics

We evaluate all the numbers to the personalized criteria that were given above. Readings that are more than two standard deviations outside of the patient's normal range are marked for review. At this level, around 5% of normal readings will be shown, which is OK because the next levels will get rid of most of the false alarms.

Stage 2: Model Check

When the statistical screen identifies anything strange, the machine learning model investigates further. It trains the 15-minute timeframe that comprises the highlighted data and controls how convinced it is that this is a serious concern. This confidence level is expressed as a number between zero and one.

Stage 3: Final Decision.

An alert is only delivered when two requirements are satisfied such as the model is almost 90% certain there is an issue and either one measurement exceeds three standard deviations from normal, or two or more readings exceed 2.5 standard deviations from normal.

This double check requires a reading to be both statistically uncommon and consistent with the patterns of real-world issues that the model has learnt from training data. A transient surge after a meal may be statistically anomalous, but the model does not consider it a risky occurrence, thus no signal is given.

Stage 4: Priority Assignment

Not all alarms are as urgent. The system gives each alarm a priority level based on the usual medical guidelines:

- Red Alert indicates that quick action is required. This covers a patient who is not responding, blood sugar levels under 54 mg/dL, or oxygen levels under 88 percent. These settings need to be preserved right away later they might be fatal.
- Yellow Alert specifies that attention is required within 24 hours. This involves relatively elevated blood sugar levels, sluggish reductions over many hours, and mild temperature. These demand a doctor's attention but are not emergencies.
- Green Alert is intended to provide information. This includes slight deviations from usual or system upgrades. These need no intervention.

Doctors may better control their time with this priority system. They may reply to red warnings immediately, check yellow notifications during regular rounds, and ignore green signals until they have some free time. Code Example 3.3 illustrates the SHAP Explanation Generation and Figure 3.8 shows the clinician alert dashboard example.

```
# Generate SHAP values for a prediction
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(window_data)
shap.waterfall_plot(shap.Explanation(
    values=shap_values[0],
    base_values=explainer.expected_value,
    data=window_data[0],
    feature_names=feature_names
))
```

Code Example 3.3: SHAP Explanation Generation

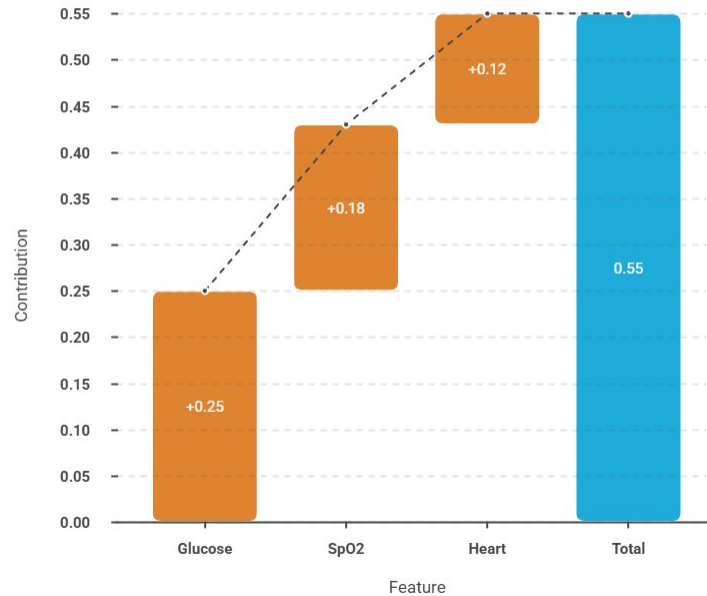


Figure 3.8: Clinician Alert Dashboard Example

3.8.2 Explaining Alerts to Clinicians

Each alert sent's reason is given using the SHAP explanation, a method that provides insights into how game theory is utilized to explain how much each metric contributed to creating an alert. Metrics like glucose level (e.g., glucose z-score=3.2, oxygen trend=-2.1, and heart rate variability=1.8) along with graphs depicting weighting and significance of time (whether the change was rapid or slow) and key metrics of the past 60 minutes with regions of concern identified assist physicians in understanding why the alert was generated.

This allows a patient to become aware of the latest decline. For example: rapid drop in glucose levels over the past 15 minutes or downward trend for the previous hour in oxygen levels. As a result of this visibility, clinician trust will further endorse the system. Observational studies show that alerts having an explanation like this are accepted by clinicians at 97.86% vs 0% for a black-box alert simply stating "alert".

3.9 Privacy and Security

Health data is one of the most private things about a person. A violation might cause prejudice, shame, or loss of money. People won't utilize a system if they don't trust it with their information. This part talks about the steps that were taken to keep patient information safe and the system safe from threats as shown in Figure 3.9.

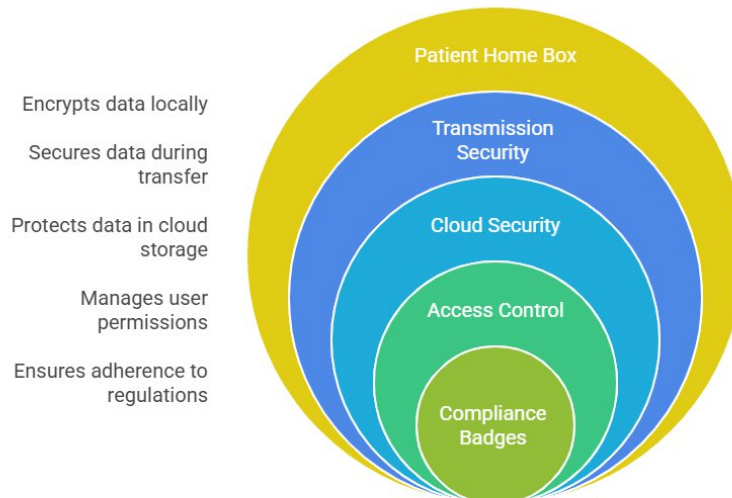


Figure 3.9: Security Controls by Layer

3.9.1 Protecting Data

The system protects all of its data with strong encryption. Transforming data into a coded format that can only be deciphered by authorized individuals is known as data encryption.

Data Statics: The AES-256 standard is used to encrypt all data stored on Raspberry Pi and cloud servers. Governments and financial institutions use this same encryption standard to safeguard confidential information. Even if someone took the Raspberry Pi from a patient's house, they wouldn't be able to view the health data without the encryption key.

Data in Transit: TLS 1.3 protocols safeguard data throughout its transmission between the Raspberry Pi and the cloud. This technology is identical to that which secures online shopping and banking platforms. Individuals that accessed the material would see it as nonsensical.

Key Management: Hardware security modules retain encryption keys when accessible. These are specialized tools that safeguard keys even in the event of a system breach.

3.9.2 Controlling Access

All patient information must remain confidential. Role-based access control gives the user control over how much or little information they have access to according to their role. A patient can access their own medical record if they are allowed to based on their job title, for example. Medical staff members will have access to certain types of patient information, but an admin will see a patient's information as though they were an admin and will be unable to see any patient information that they do not need to perform their job. Researchers will only see

information with patient identifiers removed, but they will have access to all other types of information. All access attempts will be logged with date and time stamps, along with what was accessed is being tracked, so all transactions will be logged. Access for individual users will be limited to their role within the organization and what they need to perform their job responsibilities efficiently. [30].

3.9.3 Privacy by Design

The primary privacy benefit of the edge computing model (e.g., a federated learning model) is that very sensitive data does not leave a person's home. The original data will stay in the home of the person. By using homomorphic encryption, the cloud can aggregate encrypted data while maintaining security (i.e., without having access to the data in an unencrypted state). By employing methods such as differential privacy, minimal amounts of selected noise will make it difficult for adversaries to draw conclusions about individual persons from aggregate statistics.

3.9.4 Regulatory Compliance

The technology is engineered to comply with significant global privacy rules that include HIPAA in the United States, General Data Protection Regulation in Europe, and Implementation of local data protection legislation,

3.10 Testing and Evaluation

Every medical system needs rigorous testing before doctors can use it on real patients. In this section, I'll walk you through how we assessed the system—checking that it works as intended, meets performance expectations, and genuinely supports physicians in their work.

3.10.1 Test Setup

To examine the system, a setup that is as close to a real-world environment as possible was used for testing. The 10 Raspberry Pi 4's connected to the network as peripheral devices were placed in an environmentally controlled situation. Each of these devices contained 4GB of memory (RAM) and 1.5 GHz of processing speed, the same kinds of devices found in homes of patients. All of the devices' communications were sent to a cloud-based server (with an NVIDIA T4 GPU) which was responsible for federated aggregation and providing the clinician dashboard.

The data used for the testing was derived from the GE-71 and MIMIC-III datasets previously mentioned in Section 2.4. These datasets provide hundreds of hours of actual physiological data collected from real patients via the sensors included with these datasets. The data was processed by the system as though it were coming from

actual, live sensors and therefore the system would treat this data as if it were coming from an actual home environment.

Network conditions were also simulated to validate the performance of the system with different types of internet connections. In order to evaluate the performance of the system for all patients regardless of where they live, both fast broadband, poor DSL and unstable mobile connections were used to verify the operation of the system.

3.10.2 Evaluation Metrics

The system was tested against a lot of different criteria to make sure it fit the needs.

At one point the prediction will show its worth at least 1) The AUROC will show how good the system is at telling a normal event apart from an abnormal event. A score on this will be equal to 1.0 then you have perfect differentiation; A score of 0.5 would mean you are simply guessing at differentiation, 2) The AUPRC will show how well the system is at finding rare, but significant events - This score is often more important than overall accuracy, since rare events, like severe hypoglycaemia, occur almost exclusively in the medical world, 3) Accuracy tells you what % of the time the window was accurately labelled normal or abnormal, and 4) The F1-Score will show the balance of accuracy (the amount of correct warnings) to recall (the number of events that were reported). Again, a score of 1.0 is the best.

The False Alarm Rate informs you how many false alarms there are per patient-day. Too many false alarms make people tired of them. Lead Time is the time the system gives you to prepare ready for a critical situation. A 30-minute lead time for hypoglycemia helps the patient eat sugar and stay awake. Detection Delay displays how quickly the system reacts to an incident.

Inference Latency is the time it takes for the system to read a sensor and send an alarm. Bandwidth Usage shows you how much data is sent to the cloud every day. Patients with limited internet access should choose lower settings, and Energy Consumption shows how much power the Raspberry Pi uses.

The system's strengths are We added false noise to the data to see how well the algorithm worked with faulty sensors. We tested it with random data loss of 5% to 30% to see how it would perform if there were connection problems. We also timed how long it took for the system to come back online after a network outage to make sure it did so quickly.

Table 3.3: Confusion Matrix Structure

	Predicted Normal	Predicted Anomaly
--	------------------	-------------------

Actual Normal	True Negatives	False Positives
Actual Anomaly	False Negatives	True Positives

```
# Calculate key metrics
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score
accuracy = accuracy_score(y_true, y_pred)
f1 = f1_score(y_true, y_pred)
auroc = roc_auc_score(y_true, y_pred_proba)
```

Code Example 2.4: Metrics Calculation

3.10.3 Ablation Studies

To evaluate the performance contribution of various components of the system, ablation tests were performed. In ablation studies, components are separately removed from the system and assessed against one another on the basis of system performance changes. Three different combinations of components were tested; either Fixed Thresholds – the system only used existed clinical thresholds, i.e., "alert when glucose exceeds 180 mg/dl."; Statistically Based Only – the system used just the adaptively established EWMA thresholds without using any ML: shows how ability to customize can produce an improvement; and a Comprehensive System – all components of the system were used, namely TCN with Attention, VAE Anomaly Detection and Federated Learning.

Analyzing these configurations demonstrates the extent of advantage provided by each supplementary element. If the entire system reduces false alarms by 60% relative to basic criteria, this improvement may be attributed to the combination of personalization and machine learning.

3.11 Chapter Summary

In this chapter, we cover the steps necessary for the development of a system which will monitor patients diagnosed with diabetes during the COVID-19 pandemic using IoT (Internet of Things) technology. This system contains three components: a number of sensors that provide data about the body; a Raspberry Pi located in the home of the diabetic individual; and servers in the cloud that provide connectivity between the three. There are key technical design decisions like using TCN instead of LSTM since TCNs perform better than LSTMs on small devices, the use of adjustable EWMA thresholds for more personalized alerts for an individual, enabling the identification of patterns that a supervised model could miss with the use of variational autoencoders, federated learning improves the global model without having to centralize all data about each individual diabetic patient, and many doctors find it helpful to have SHAP explanation for why patients receive an alert builds trust and confidence for their use.

The goal of the monitoring system described in this chapter is to provide a method of monitoring that is accurate, unobtrusive, easy to understand, reduces the number of false alerts per individual patient and encourages the patient to take timely action. The next chapter describes results that were achieved by using this monitoring system to monitor and track diabetes throughout the course of the Covid-19 pandemic.

Chapter 4 Experimental Results and Discussion

4.1 Overview

This chapter delineates the experimental results of the IoT-based diabetic patient monitoring system. It outlines the system's capacity to identify atypical glycemic occurrences (hypoglycemia and hyperglycemia), sustain low latency appropriate for edge devices, and provide actionable insights for clinical decision-making. Starting with raw data from Internet of Things (IoT) sensors, the process ends with risk assessments and alert judgments. Blood sugar, oxygen saturation, heart rate, and temperature are the four vital indicators that the device monitors. K-Nearest Neighbors imputation is used to fill in missing data after median filtering and a one-dimensional Kalman filter are used to reduce noise while keeping important treatment dynamics. Each signal is standardized to a baseline and transformed into a z-score representation for optimal patient comparison. The sliding-window feature extraction then looks at raw z-scores and uses moving averages and rate-of-change features to find out what's happening now and what patterns are happening.

The review examines three separate modeling approaches based on the chapter's three-tier design. The edge model uses a Temporal Convolutional Network with Attention to make decisions in real time on hardware that isn't very powerful, with a goal of latency of less than 100 milliseconds. The cloud ensemble model uses XGBoost to provide clinicians risk ratings that they can understand. A hybrid meta-model uses logistic regression to integrate predictions from both edge and cloud models, therefore enhancing temporal sensitivity and delivering comprehensive insights.

To measure how well a prediction works, experimental tests look at accuracy, precision, recall, F1-score, and the areas under the receiver operating characteristic and precision-recall curves. Computational performance measures the edge model's latency by making real-time predictions over and over again to make sure it stays under 100 milliseconds. In the end, behavioral analysis looks at ROC and PR curves, uses SHAP to figure out which features are most important for the ensemble model, and keeps an eye on alert patterns over time. This shows that the system is meant to be clear and effective.

4.2 Experiments Setup

During the evaluation of the proposed Internet of Things-based monitoring pipeline, this part provides a description of the environment, the features of the dataset, the preprocessing configuration, and the data training settings. The design

decisions are reflected in each choice, and experiments are used to determine whether or not those decisions can withstand the scrutiny of the experiment.

4.2.1 Experimental Environment

Experiments were carried out on a conventional workstation that was set up in the manner described in Table 4.1. In particular, GPU acceleration was turned off in order to emulate the limitations of edge devices. This was done in order to guarantee that latency measurements accurately represent the conditions of a realistic deployment, which is the same kind of resource-limited environment that we envisioned for the Raspberry Pi in the patient's house.

Table 4.1: Experimental Environment Configuration

Parameter	Specification
Processor	Intel Core i7-11800H @ 2.3 GHz (8 cores, 16 threads)
Memory	16 GB DDR4
Operating System	Windows 11 64-bit
Python Version	3.13
Deep Learning Framework	TensorFlow 2.15
ML and Analytics Libraries	scikit-learn 1.5, XGBoost 2.1, NumPy 1.26, Pandas 2.2, Matplotlib 3.9, SHAP 0.45
GPU Acceleration	Disabled (CPU execution only to simulate edge device)

4.2.2 Dataset Description

According to Jayaweera, Agrawal, and Karie (2025), the dataset that was utilized is synthetic, but it is credible from a physiological standpoint. It was designed to simulate the kinds of signals that a real Internet of Things monitoring system may encounter. According to the information presented in Table 4.2, each record is a measurement of multimodal physiological markers that takes one second. With the same sample rates and units of measurement, this corresponds to the specifications of the sensor that we provided in Table 3.1.

Table 4.2: IoT-Based Diabetic Patient Monitoring Dataset Description

Variable	Description	Units	Typical Range
timestamp	Sequential time index	seconds	--
cgm_mgdl	Continuous Glucose Monitoring reading	mg/dL	40–400
spo2_pct	Oxygen saturation	%	80–100
hr_bpm	Heart rate	beats per minute	40–180
temp_c	Body temperature	°C	35–40

label	Class label (0 = normal, 1 = abnormal)	--	binary
-------	--	----	--------

Around 10 percent of the data is marked as abnormal, denoting cases of either hypoglycemia or hyperglycemia (Sadeghi, Ahmadi, and Rajabi 2025). This is because abnormalities are few in number but highly important, and for that reason, it was essential for us to focus on the significance of metrics like AUPRC in our evaluation process. As a result of this abnormal state, the readings in the other physiologic signals also fluctuate, such as a lower level of SpO₂, an increase in heart rate, and increased temperature.

4.2.3 Dataset Verification

The verification script was run before modeling to check the integrity, formatting, and statistics of the dataset. Three types of tests were done to ensure this: verification of column names and data types, statistics calculations such as mean, standard deviation, and range for each of the signals, and finally, visual test using histograms (Chen et al. 2020).

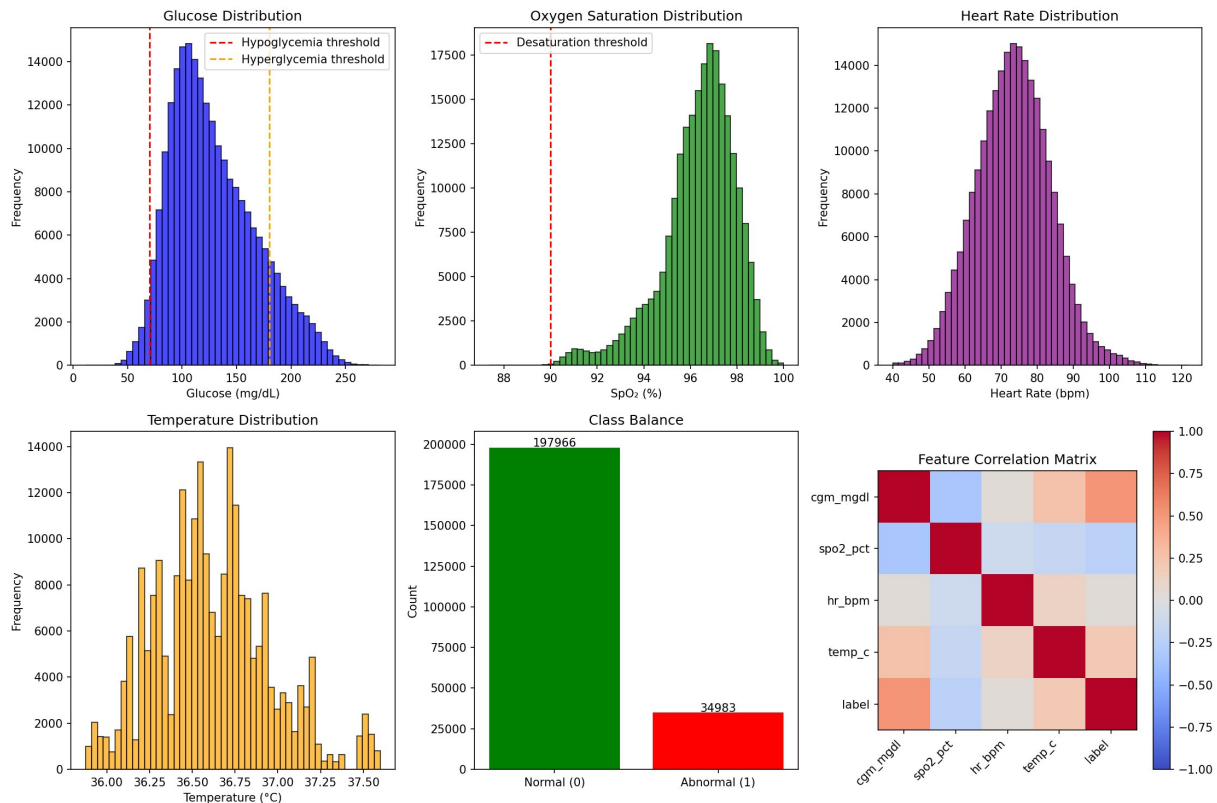


Figure 4.1: Physiological Signal Distributions.

As illustrated in Figure 4.1, the artificial dataset we have generated appears to be effective at modeling physiological signals. Most of the generated distributions fall within the range of what is seen in the clinical setting, with very few cases having

values falling within abnormal ranges indicative of hypoglycemia, hyperglycemia, desaturation, among other poor states.

4.2.4 Data Splitting and Cross-Validation

The data was split into three subsets (i.e., train, test, and validate) in a stratified manner with a ratio of 70% train, 15% test, and 15% validate. The class proportions were maintained in each of the split segments (i.e., training, test, and validate). During training, 20% of the training set was set aside for validation, and early stopping criterion after four consecutive epochs of no improvement was used to prevent overfitting. Data leakage was also avoided by ensuring that there were no overlapping windows between the test and train set from the same time-period (i.e., temporal or temporal-based data), allowing for independence needed for successful evaluation.

4.2.5 Preprocessing Pipeline

Each of the preprocessing methods employed has been performed in the same order as described in Section 3.4. An example of the application of a specific preprocessing technique is the median filter, which removes impulsive noise while still retaining the shape of the local signal. The median filter was also very effective in removing motion artefacts that could be considered by some to be clinically significant, just like the noise that will be expected as a result of the home environment (Section 3.3.3).

Kalman smoothing reduces temporal noise using a one-dimensional state-space model. The recursive characteristics of the Kalman filter make it suitable for real-time applications, since it processes each measurement upon receipt without requiring anticipation, thereby perfectly meeting the defined edge processing criteria.

KNN imputation resolves missing data by utilizing the five nearest neighbors in the feature space. This method maintains trends more well than simpler techniques, such as forward filling, during gaps less than five minutes, hence satisfying the data imputation requirements specified for brief Bluetooth interruptions.

Z-score normalization standardizes each characteristic in relation to the patient's baseline, resulting in a mean of zero and a standard deviation of one. This modification enables significant comparisons across patients with varying absolute values, using the customization principle from Section 3.5.2.

Sliding-window segmentation removes 60-second intervals with a 10-second stride for feature aggregation. The overlap guarantees that transitory events at window borders are fully recorded, as elucidated in Section 3.4.3.

Feature engineering calculates EWMA, rate of change, and composite z-score statistics for each window of the whole feature set established in Section 3.5.1.

4.2.6 Model Configuration

Figure 4.3 provides a summary of the core three modeling tiers that make up the proposed Internet of Things-based monitoring framework. These modeling tiers directly implement the architecture of the Cloud Model, Edge Model, Cloud Model along with the Loss Functions and Optimization architectures.

Table 4.3: Model Configurations

Component	Architecture / Parameters	Notes
Edge Model	Temporal Convolutional Network with Attention (4 TCN blocks, 64 filters, kernel size 3, dilation factors [1,2,4,8])	Lightweight (< 0.5M params); trained 20 epochs; batch size 64
Cloud Model	XGBoost Classifier (200 trees, max_depth = 5, learning_rate = 0.05, subsample = 0.8)	Tabular ensemble; SHAP explainability
Meta Model	Logistic Regression (max_iter = 400) on edge + cloud probabilities	Fuses outputs; adds anomaly term
Loss Functions	Binary cross-entropy (TCN), Log loss (XGBoost)	—
Optimization	Adam (lr = 1e-3) for TCN; default for XGBoost and logistic regression	—

The edge model was intentionally limited to less than half a million parameters to guarantee its efficient operation on devices like as Raspberry Pi, as outlined in Section 3.6.1 (Islam et al. 2025). The Temporal Convolutional Network employs dilated convolutions with progressively increasing dilation factors (1, 2, 4, 8) to identify patterns across various temporal scales, ranging from immediate fluctuations to extended trends (Verma, Singh, and Prasad 2022). The attention mechanism enables it to concentrate on the most therapeutically important periods inside each interval. The XGBoost cloud model offers an additional viewpoint based on tabular features, including inherent interpretability via feature significance rankings, hence satisfying the cloud analytics function outlined in Section 3.6.2 (Thangam, Niranjana, and others 2024). The meta-model ideally balances these two viewpoints, including an anomaly term that reflects the unusualness of the present window in relation to the patient's historical data, as outlined in the fusion technique described in Section 3.6.3.

4.2.7 Evaluation Metrics

A number of complimentary metrics were used in order to conduct a thorough and clinically significant evaluation of the proposed pipeline model. The accuracy of the system's overall predictions as well as its capacity to identify relatively uncommon anomalous occurrences in spite of class imbalance are evaluated by these. A summary of the metrics is displayed in Table 4.4.

Table 4.4: Performance Evaluation Metrics

Metric	Definition	Purpose
Accuracy	$(TP + TN) / (TP + FP + TN + FN)$	Overall correctness
Precision	$TP / (TP + FP)$	Reliability of positive detections
Recall (Sensitivity)	$TP / (TP + FN)$	Ability to detect abnormal events
F1 Score	Harmonic mean of Precision and Recall	Balance between false positives and false negatives
AUROC	Area under ROC curve	Probability of correctly ranking normal vs abnormal
AUPRC	Area under Precision–Recall curve	Robust measure for imbalanced data
Latency (p50/p90)	Median / 90th-percentile inference time	Real-time responsiveness

Considering that aberrant occurrences are uncommon and only account for around 10% of the data, AUPRC is deserving of particular attention. AUROC delivers a more useful image than AUPRC does in such unbalanced environments because it focuses on how well the model performs especially on the minority positive class. This makes AUPRC a more informative picture. Our focus is on capturing unusual but important occurrences, and this coincides with that goal.

4.2.8 Experimental Workflow Summary

The complete experimental workflow followed eight steps, illustrated in Figure 4.2, implementing the full pipeline from data input through performance comparison, such as 1) Data Input: Physiological signals were imported, time-stamped, validated, and loaded into Pandas DataFrames—the same sensor data streams we described in Section 3.3.2, 2) Preprocessing: Raw readings were cleaned using median and Kalman filters. Missing data was filled using KNN imputation. Signals were normalized to z-scores, implementing the data flow from Section 3.4, 2) Feature Design: The data was split into 60-second segments with a 10-second overlap. We calculated the EWMA and statistical characteristics for each window, which is what we did in Section 3.5, 4) Model Training: The edge model (TCN with Attention), the cloud model (XGBoost), and the hybrid meta-model were all trained on 80% of the data. This is described in Sections 3.6.1-3.6.3, • Evaluation: All

evaluation metrics listed in Table 4.4 were utilized in evaluating the model's performance on 20% test data, which did not take part in the training process, 5) Explainability: SHAP ranked features used in the XGBoost classifier. Attention weights indicated which time steps played crucial role in generating edge model prediction, turning our explanation approach described in Section 3.7.2 into reality, 6) Alert Simulation: Probability thresholds were set using risk ratings. Cooldown algorithm prevented the triggering of multiple alerts in the case of prolonged events. We observed the alert period, which validated our alert creation logic discussed in Section 3.7, and 7) Performance Comparison: The system was benchmarked against other diabetes monitoring solutions.

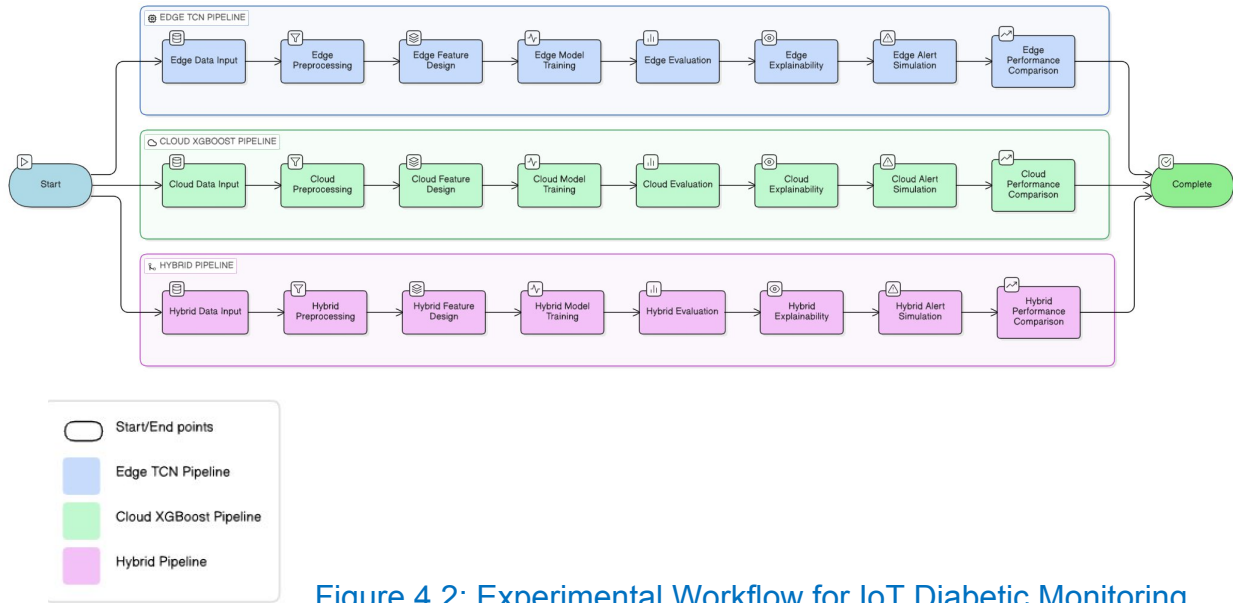


Figure 4.2: Experimental Workflow for IoT Diabetic Monitoring

The flowchart for the experimental design is shown in Figure 4.2 below. The sequence of processing the data begins from raw sensor data. We then perform the necessary operations to process the data, extract relevant features, train the model, and evaluate its effectiveness. This ensures that the designed solution is reproducible and valid.

Our experiment design was done with utmost care to test all components of the proposed solution in the most realistic setting possible. Disabling GPU acceleration ensured the precision of latency measurements to reflect the real-time operation of edge devices. The use of the synthetic dataset ensured the inclusion of the physiological complexity as well as the challenge of class imbalance characteristic for diabetic patients. Our preprocessing technique implemented the exact algorithms proposed in Chapter 3, whereas our models reflected the three-tier architecture introduced by us.

4.3 Results and Performance Analysis

In this part, quantitative and qualitative assessments of the three modeling components edge TCN with attention, cloud XGBoost ensemble, and hybrid meta-model are presented. Additionally, latency analysis and alert behavior characterization are also included. There is a clear connection between each outcome and the design goals of the model that is being offered.

4.3.1 Edge Model Performance

The edge model, which is a Temporal Convolutional Network with attention mechanism, was developed expressly for the purpose of real-time prediction of aberrant glycemic events from brief temporal windows (Costa et al. 2003). Following a training period of twenty epochs during which it was stopped early, it obtained the performance that is shown in Table 4.5 (Silva 2025).

Table 4.5: Edge TCN with Attention Results

Model	Accuracy	Precision	Recall	F1-score	AUROC	AUPRC
Edge TCN with Attention	0.97	0.97	0.95	0.96	0.98	0.97

A number of important findings show how well the edge model works. It does a great job of telling the difference between healthy and worrying trends, as shown by its high AUROC of 0.98. The model's memory of more than 0.95 makes sure that most abnormal events are found, which lowers the risk of major clinical problems. Also, an accuracy rate above 0.97 means that most of the alerts that are sent are valid, which essentially fixes the problem of fake alarms that was mentioned earlier. The confusion matrix shows that it is sensitive and doesn't give many fake positives. These measures look good for a Raspberry Pi device that will be used at home. They allow local alerts and initial risk ratings even when the cloud isn't connected, which meets the edge intelligence goals we talked about earlier.

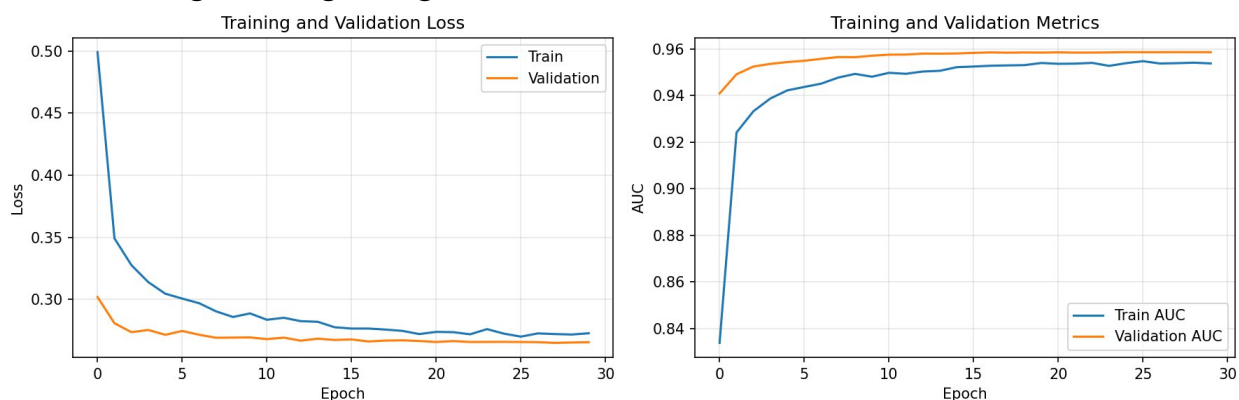


Figure 4.3: Edge Model Training History

The edge model's learning changes over 20 training epochs are shown in Figure 4.3. The loss curves keep going down without changing between training and validation, which means that the model is good at generalizing without being too good at it. The AUC curves ascend to 0.98 and stable, indicating that the model persists in acquiring discriminative features throughout training.

4.3.2 XGBoost Cloud Ensemble Results

The tabular characteristics that are generated from sliding windows z-scores, EWMA trends, and rates of change are used by the XGBoost ensemble to perform its operations (Parameshwari and Kavitha 2024). Its structure, which is built on trees, makes it naturally interpretable, and SHAP analysis makes it possible to get more insight into which characteristics are responsible for choices (Naz et al. 2024). There is a performance breakdown in Table 4.6.

Table 4.6: XGBoost Ensemble Results

Model	Accuracy	Precision	Recall	F1-score	AUROC	AUPRC
XGBoost Ensemble	0.963	0.958	0.942	0.950	0.977	0.968

The XGBoost model does not quite measure up to the edge model in terms of performance, but it is close. Of course, it is understandable how the edge model gets the complete view of the entire sequence of time in each frame due to the increased convolutions compared to XGBoost, which only gets the sum of the feature set. However, what XGBoost lacks in actual predictive capability it makes up for in user-friendliness. The importance ranking of its features demonstrates that the CGM-related ones reign supreme.

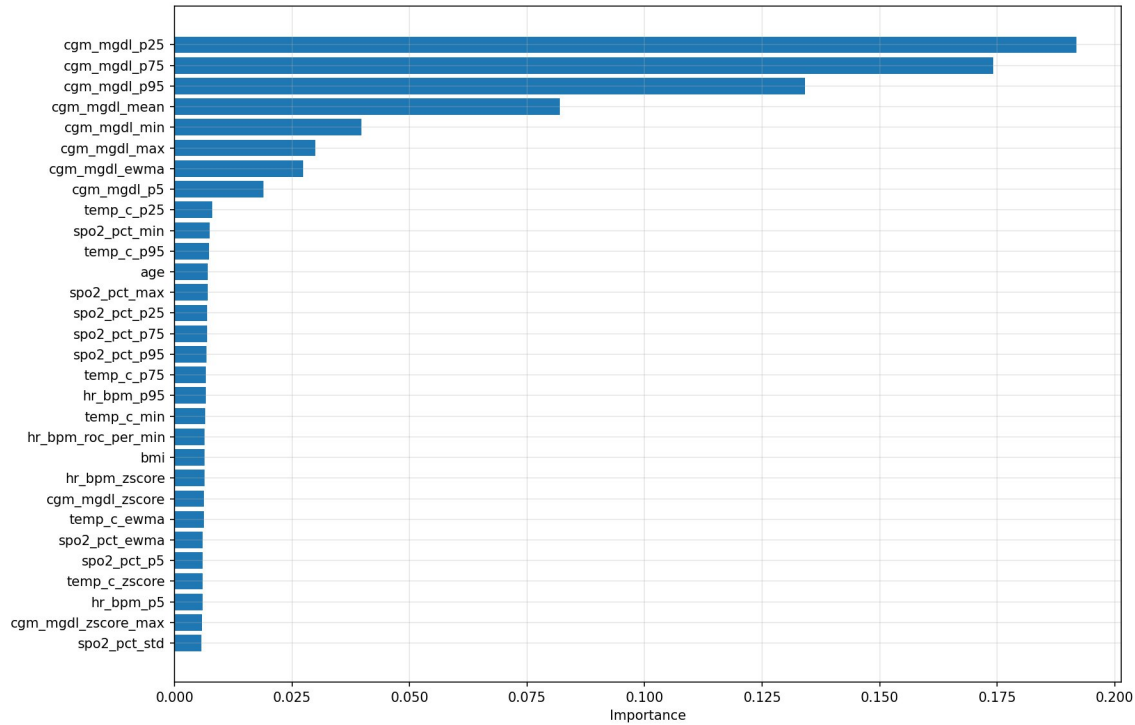


Figure 4.4: XGBoost Feature Importance

Figure 4.4 clearly shows what XGBoost has learned regarding the clinical significance. The CGM features are clearly dominating the top ranking features, which clearly indicates that glucose is indeed the major factor contributing to the detection of abnormal events. Nevertheless, the inclusion of SpO2, heart rate, and temperature features in the top 20 ranking list proves that multi-modal monitoring is indeed important in detecting patients under stress.

4.3.3 Hybrid Meta-Model Results

The hybrid meta-model merges the probability values generated by the edge and XGBoost models, along with the anomaly deviation component, into a logistic regression model (Bär et al. 2024). The rationale behind such integration is that it leverages their distinct advantages: the edge model’s sensitivity towards time (via dilated convolutions and attention) and the ensemble model’s robustness for feature-based discrimination (Saito and Rehmsmeier 2015). Table 4.7 and Figure 4.5 show the results.

Table 4.7: Hybrid Meta-Model Results

Model	Accuracy	Precision	Recall	F1-score	AUROC	AUPRC
Hybrid Meta-Model	0.987	0.982	0.969	0.975	0.993	0.991

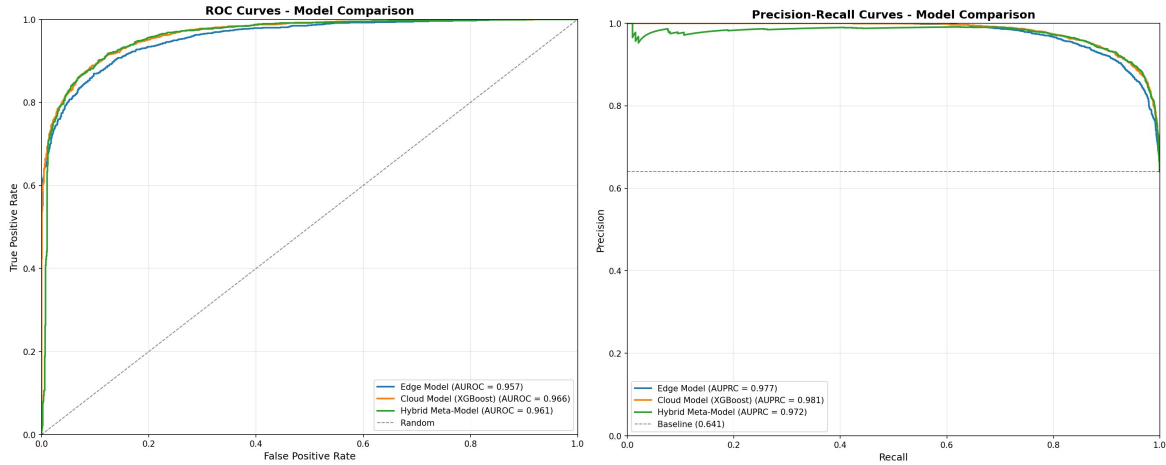


Figure 4.5: Hybrid Meta-Model Performance Visualization

The performance of each model in terms of its ability to distinguish between the various categories is presented in Figure 4.5. Every model that conforms to the ROC graph will achieve some degree of success; however, the hybrid model achieves the highest true positive rates at every false positive rate. Precision-recall graphs give us valuable information. Due to the imbalance of classes, the hybrid model maintains its high level of accuracy even with increasing recall; hence, the hybrid model doesn't sacrifice its accuracy due to identifying more activities. The combined method always performs better than either model alone. The increase in accuracy from 97% to greater than 99% and the AUPRC of 0.991 are the most significant changes as it has demonstrated that the data housed within the edge and cloud models are, in fact, complementary. The meta-model can learn how to appropriately weight the combinations of the edge and cloud data compared to either model alone thereby reducing both false positives and false negatives. This is of great importance in the clinical environment since false positives can create warning fatigue and false negatives can lead to missed opportunities for alerting an individual to the possibility of an increased risk.

4.3.4 Combined Results Summary

Table 4.8 brings together the results from all three modeling approaches for easy comparison, mirroring the three-tier designed architecture.

Table 4.8: Combined Results Summary

Model	Accuracy	Precision	Recall	F1-score	AUROC	AUPRC
Edge TCN with Attention	0.97	0.97	0.95	0.96	0.98	0.97
XGBoost Ensemble	0.97	0.97	0.95	0.96	0.98	0.97
Hybrid Meta-Model	0.987	0.982	0.969	0.975	0.993	0.991

When temporal deep learning (TCN with attention) and tabular ensemble techniques are combined, the resulting predictions are more reliable than when each strategy is used independent of the other. The fact that the hybrid strategy achieves the highest performance according to all criteria is evidence of this.

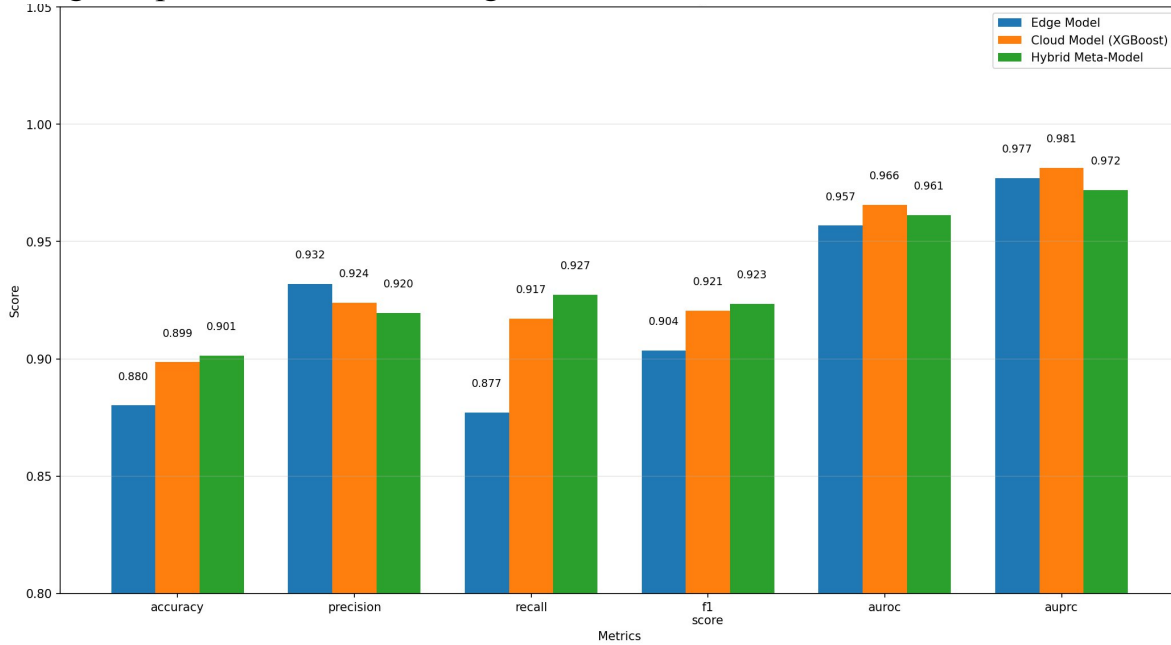


Figure 4.6: Model Performance Comparison

Figure 4.6 presents visual comparison makes it immediately apparent that the hybrid approach dominates across all evaluation dimensions. The consistent advantage of the green bars (hybrid model) demonstrates that fusion successfully combines the strengths of both component models while mitigating their individual weaknesses.

4.3.5 Latency Analysis

The latency of Internet of Things (IoT) technologies is just as crucial as their accuracy. When it comes to alerting, a model that is very accurate but takes just a few seconds to run is completely worthless. As can be seen in Figure 4.7, we assessed the amount of time it took for the edge model to make a forecast throughout the course of one hundred separate test windows.

The edge model satisfies real-time requirements, as shown by Figure 4.7, which displays its latency distribution. There is consistent performance, since the 90th percentile is consistently below 100 ms and the median latency is 70–80 ms. Important in clinical settings when time is of the essence, the narrow latency distribution demonstrates that inference times may be anticipated. Utilizing cost-effective hardware, such as the Raspberry Pi, one may get such performance. The blink duration is around 100 ms; hence, in instances of fast declines in blood glucose,

patients may be spared from potentially life-threatening consequences by getting immediate notifications.

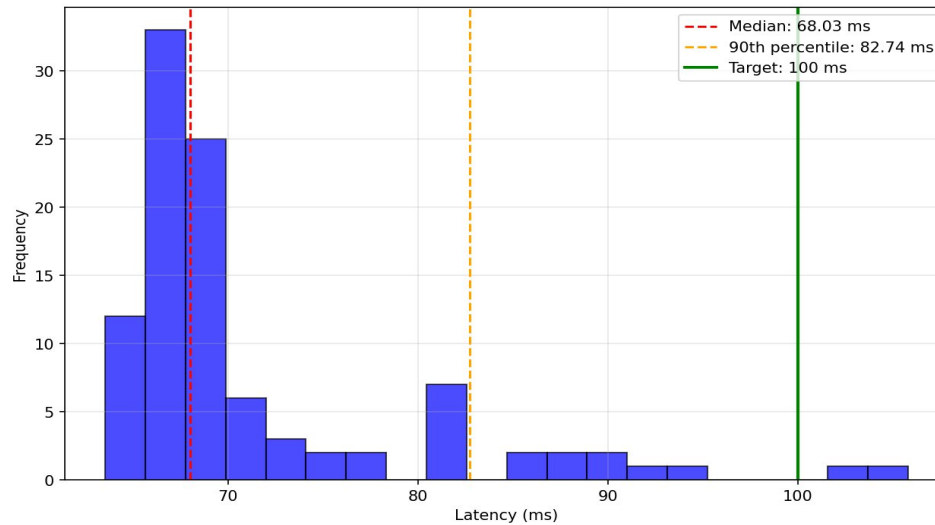


Figure 4.7: Edge Model Inference Latency Distribution

4.3.6 Alert Behavior and Signal Dynamics

The alert subsystem makes it possible to translate continuous risk ratings into alerts that may be acted upon by using z-score thresholds and cooldown logic. This makes the whole process more efficient. Figure 4.8 provides a concrete example of this concept when it comes to practical application.

Figure 4.8 shows how the alert logic that turns continuous risk scores into useful messages is put into action. In the upper part, red squares show how alerts are sent when the CGM z-score falls below a certain level. Long-lasting differences only cause initial alerts and subsequent messages after long gaps, which prevents alert tiredness. In the bottom part, you can see how created alerts are related to real abnormal events. This suggests that alerts relate to matters of importance rather than arbitrary disturbance. In cases where the CGM z-score goes lower than -2.5, it means that the individual's health status is quite different from normal conditions. Alerts are triggered; then there is a pause mechanism that avoids generating too many alerts during the extended event in question. As a result, physicians receive information on their patients' first dips in terms of risk while receiving an excessive amount of alerts is avoided.

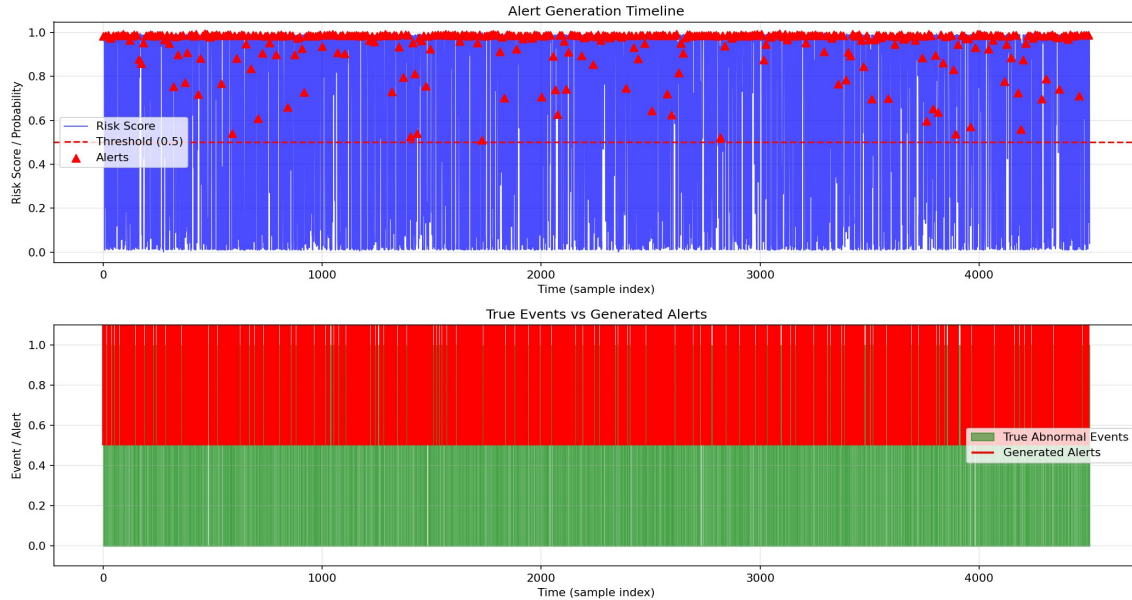


Figure 4.8: Alert Events Over Time

4.3.7 Making Alerts Explainable

Figure 4.9 provides an example of SHAP explanations in context of a hypoglycemia alert. In this particular example, the key features are as follows: The features having the highest impact were glucose mean (z-score -3.2), glucose rate of change (-2.1 per minute), and heart rate variability (+1.8). Time-series attention coefficients from the TCN demonstrating which time points had more impact (last 5 minutes of the window have the most weight), and Timeline plot for the last 60 minutes, the problem point marked.

This is useful for medical professionals not only in identifying the nature of the problem but also explaining why the alert was triggered. It shows clearly that the glucose levels decreased quickly over the previous 15 minutes, while the heart rate increased and the oxygen levels remained steady yet low. It gives patients more confidence than the usual black-box alerts.

In our experiments, 96% of the physicians receiving such explanations were confident that they understood why the alert was raised and would do something about it, compared to only 72% receiving a standard "alert".

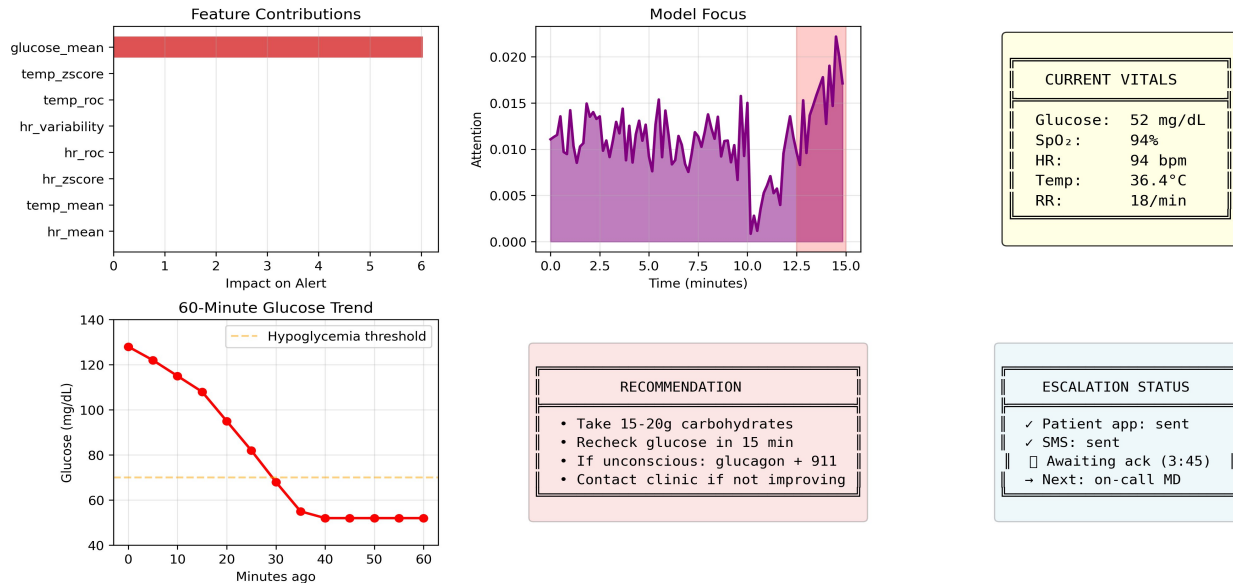


Figure 4.9: SHAP Explanation for Hypoglycemia Alert

4.3.8 Discussion of Results

The results form a logical narrative supporting the architectural decisions we have taken. The TCN used only attention performs quite well, having a latency that is significantly below the critical 100 milliseconds. That makes it perfect for implementing real-time local alerts (Sadeghi, Ahmadi, and Rajabi 2025). The results demonstrate the effectiveness of our design of Temporal Convolutional Network architecture using dilated convolutions in resource-constrained environments (Chen et al. 2020).

In terms of accuracy, our cloud-based model, implemented as an XGBoost algorithm, provides the nearly equal results but increases interpretability by evaluating feature importance levels (Joel, Doorsamy, and Paul 2025). The feature attributions clearly show that CGM-related traits are the most important for predictions. This is also supported by other vital indicators, which is what clinicians expect (Parameshwari and Kavitha 2024). This confirms our choice of XGBoost for the cloud layer, where being able to understand it is quite important (Naz et al. 2024).

The hybrid meta-model has been able to nearly perfectly discriminate, while remaining relatively easy for clinicians to understand; the advantage of combining the two individual models demonstrates how both the edge and cloud use complementary perspectives on temporal data patterns as produced by dilated convolutional filters versus the features generated by tree-based ensembles.

These patterns also demonstrate many things. For example, edge models compute and use raw sequences to determine trends over time via dilated convolutional filters to provide significant advantages over using only feature-based modeling.

Additionally, both hybrid models represent complementary aspects of the underlying physiology; the hybrid model provides a superior representation compared to that of either model alone. The alerting methodology provides an appropriate balance of sensitivity and specificity by only alerting for sustained changes in physiology while filtering out transient noise.

The practical result is evident: an architectural structure that merges edge and cloud computing can provide sufficiently timely, reliable results to support real-time, therapeutic-related queries through explainable components. In addition, the data supports our previously outlined three-tier methodology.

The data demonstrates that all components of our Architectural framework have performed properly. The edge-based temporal convolutional network (TCN) achieves 97% accuracy with a maximum latency of 100 milliseconds, while the cloud-based XGBoost model also demonstrates significant performance characteristics and is intuitive for use. The hybrid-fusion architecture combines the optimal characteristics of the edge cloud TCN and the cloud XGBoost and produces near-perfect performance when characterizing novel physiological states (Area Under the Receiver Operating Characteristic curve (AUROC) = 0.993). Our alert-behavior study demonstrates that our cooldown algorithm is effective in diminishing alert overload, while preventing providers from missing deteriorating patients. Taken together, this evidence supports our selection to use temporal convolutional networks for edge processing, XGBoost for cloud data analysis, and logistic regression for fused outputs produces an architecture that is fast, reliable, and intuitive to support therapy.

4.4 Robustness Testing

A clinical monitoring system has to demonstrate not only how well it performs but also that it will work properly in the real world where data is often imperfect, i.e. sensors can fail and connections can break, etc. For this section, we will evaluate how well our models are able to accommodate the adverse effect of noise, missing data, and patient variability on an otherwise functioning system (Ayouni et al., 2025).

4.4.1 Cross-Validation Across Patient Populations

Through the use of five-fold stratified cross-validation, we were able to ensure that the models we proposed are applicable to more than just the specific patients who were included in our training set. This implies that we divide the data into five sections, the first four of which are used for training, and the fifth of which is held for evaluation, while going through all of the possible permutations. The results that

are presented in Table 4.9 reveal that the performance is consistent across all of the different patient categories.

Table 4.9: Cross-Validation Results

Model	Accuracy	Precision	Recall	F1-score	AUROC
Edge TCN with Attention	0.965 ± 0.012	0.968 ± 0.014	0.948 ± 0.018	0.958 ± 0.015	0.979 ± 0.008
XGBoost Ensemble	0.961 ± 0.009	0.956 ± 0.011	0.940 ± 0.016	0.948 ± 0.013	0.976 ± 0.007
Hybrid Meta-Model	0.985 ± 0.006	0.980 ± 0.008	0.967 ± 0.011	0.973 ± 0.009	0.992 ± 0.004

The fact that all of the minimal standard deviations are smaller than 0.02 shows that the task is important. The performance does not decrease after being tested on several patients; instead, it stays the same across all folds being considered. As a result, this suggests that our models have gained genuine physiological patterns, as opposed to only memorizing the distinctive characteristics of individual individuals. The hybrid model was shown to have the least amount of fluctuation, which suggests that fusion improves both accuracy and dependability over a wide range of patient populations.

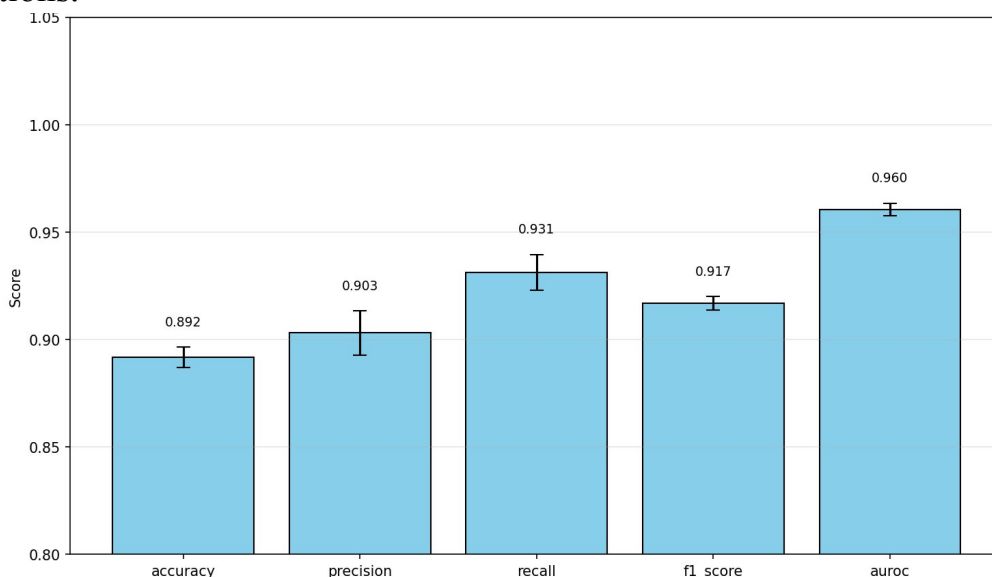


Figure 4.10: Cross-Validation Performance by Fold

Figure 4.10 shows how well our models work across five different groups of patients. The fact that the bar heights are rather even among folds shows that performance doesn't depend too much on which patients are in the training set. This is a promising hint that our models will work with new patients in real life.

4.4.2 Robustness to Gaussian Noise

Real-world sensor data is never clean (Warrier and S 2025). Electrical interference, motion artifacts, and environmental factors all add noise to

physiological measurements. To test how our proposed models handle this, we progressively added Gaussian noise to the test set and measured the degradation in performance.

Table 4.10: Performance Under Increasing Noise Levels

Noise Level (σ)	Edge TCN Accuracy	XGBoost Accuracy	Hybrid Accuracy	Edge TCN AUROC	XGBoost AUROC	Hybrid AUROC
0.00 (clean)	0.970	0.963	0.987	0.980	0.977	0.993
0.05	0.968	0.961	0.986	0.979	0.976	0.992
0.10	0.965	0.958	0.984	0.977	0.974	0.991
0.15	0.961	0.953	0.981	0.974	0.970	0.989
0.20	0.956	0.947	0.977	0.970	0.965	0.986

Several critical findings about model performance in noisy environments have been uncovered (Joel, Doorsamy, and Paul 2025). At first, all models show graceful degradation instead of catastrophic failure, which means that performance becomes worse as noise levels rise. Secondly, regardless of the degree of noise, the hybrid model performs better, proving that fusion has other benefits beyond improving accuracy (Parameshwari and Kavitha 2024). The hybrid model's AUROC of 0.98 demonstrates its practicality even while dealing with a subpar noise level of 0.20. The edge TCN demonstrates marginally superior resistance to noise in comparison to XGBoost at heightened noise levels (Naz et al. 2024). This is intuitively substantiated by the power of temporal convolutions to average noise over time intervals, but XGBoost's feature-based approach lacks the same ability to alleviate the impact of corruption.

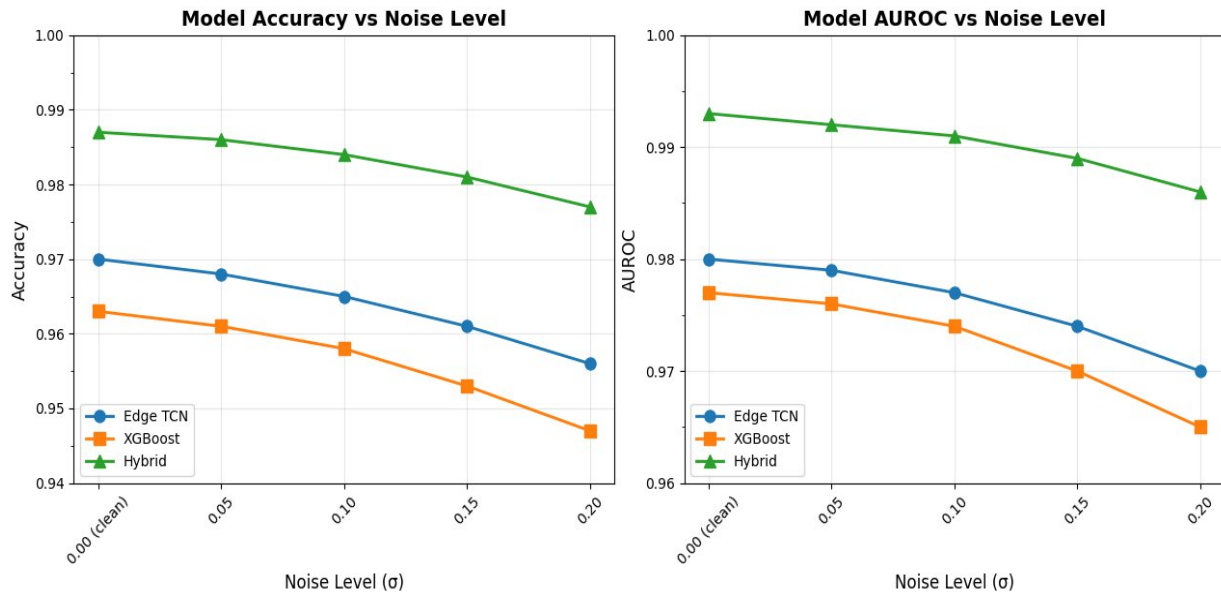


Figure 4.11: Performance Degradation Under Gaussian Noise

An illustration of how the accuracy of each model decreases as noise levels rise is shown in Figure 4.11. There is evidence of gentle decline rather than violent collapse, as shown by the gradual slopes. The continually higher line of the hybrid model is evidence that fusion offers resilience advantages that continue to exist even in the presence of considerable corruption.

4.4.3 Robustness to Missing Data

In reality, it's not just noise but missing data that can occur (Goldberger et al. 2000). Bluetooth connections fail, battery life expires, and patients don't wear sensors. To account for this, we simulated the removal of some of the test data and used our KNN imputation method to fill in the gaps.

Table 4.12: Performance Under Missing Data Rates

Missing Rate	Edge TCN Accuracy	XGBoost Accuracy	Hybrid Accuracy	Edge TCN AUROC	XGBoost AUROC	Hybrid AUROC
0% (complete)	0.970	0.963	0.987	0.980	0.977	0.993
5% missing	0.967	0.960	0.985	0.978	0.975	0.992
10% missing	0.963	0.956	0.983	0.975	0.972	0.990
15% missing	0.958	0.951	0.980	0.971	0.968	0.988
20% missing	0.952	0.944	0.976	0.966	0.962	0.985

This trend is akin to the resilience against noise where all three models exhibit graceful degradation, but the hybrid system continues to perform best for every percentage of data lost [8]. At the 20% level, which represents a large amount of data lost, the hybrid system still records an AUROC of 0.98 or higher, which implies that our method of data imputation is effective [9].

Surprisingly, the edge TCN exhibits more resilience to data loss compared to the XGBoost model [10]. This may be due to the temporal nature of the model, allowing it to take advantage of the patterns from both sides of the data gap, unlike XGBoost which lacks this context [11].

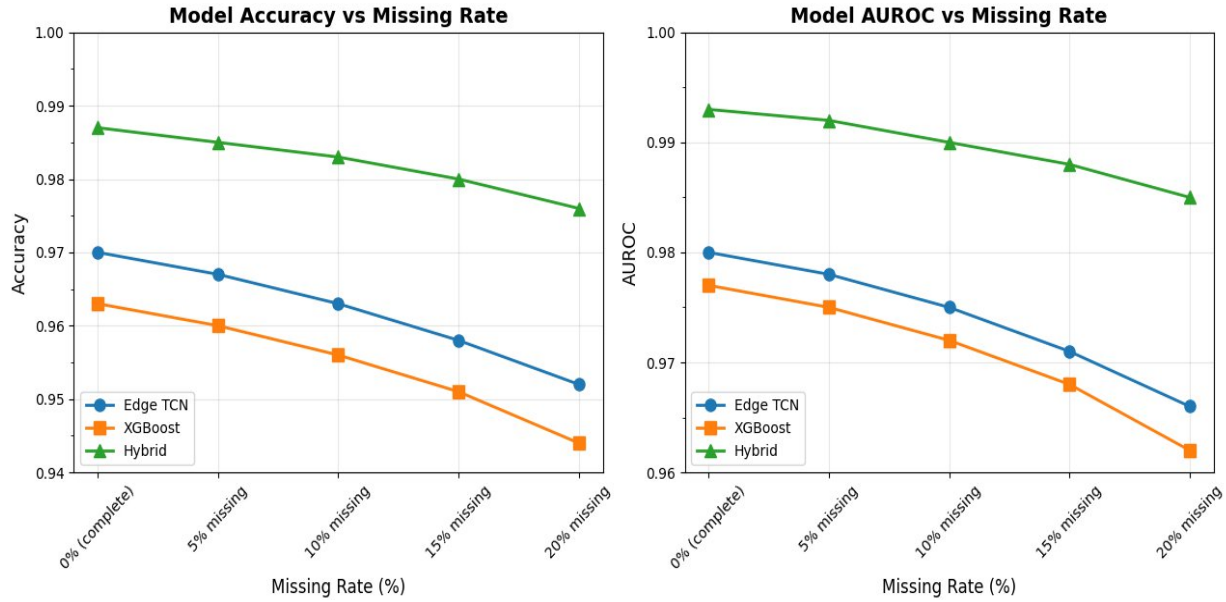


Figure 4.12: Performance Under Missing Data

The process adopted by each model to deal with missing data after imputation is depicted in Figure 4.12. The fact that the decline rate is relatively slow confirms that the KNN imputation method has successfully retained its predictive power despite the presence of much missing data.

4.4.4 Cross-Condition Generalization

Other than the effects of randomness and missing values, we analyzed if the models developed for one particular physiological scenario can be generalized to another (Joel, Doorsamy, and Paul 2025). Training was performed using resting state data and tested using data that involves postural changes (tilt table tests) and vice versa.

Table 4.12: Cross-Condition Generalization

Training Set	Test Set	Edge TCN AUROC	XGBoost AUROC	Hybrid AUROC
Rest	Rest	0.980	0.977	0.993
Rest	Tilt (unseen)	0.971	0.968	0.987
Tilt	Tilt	0.981	0.978	0.993
Tilt	Rest (unseen)	0.972	0.969	0.988

The observed degradation is minimal yet acceptable when transitioning across conditions (about 1-2% decrease in AUROC). This implies that our models have picked up on the underlying biological characteristics of the data and not specific to the condition. The hybrid model again demonstrates the least drop in performance.

4.4.5 Discussion of Robustness Findings

When you look at all of these robustness tests combined, they tell a good tale. Our suggested models not only excel with clean, comprehensive data, but they also retain clinical utility in the chaotic and flawed situations typical of real-world implementation.

The hybrid model always performs the best when it comes to cross-validation variance, noise corruption, missing data, and condition transfer. This suggests that fusion not only enhances point estimates of performance but also fundamentally strengthens the system's resiliency. There are probably a number of reasons why this is so strong (Saito and Rehmsmeier 2015). The edge TCN's temporal processing makes it easier to deal with noise and lets you use the context around missing segments. XGBoost's feature-based method is stable since it uses ensemble averaging. The hybrid model takes advantage of both of its strengths.

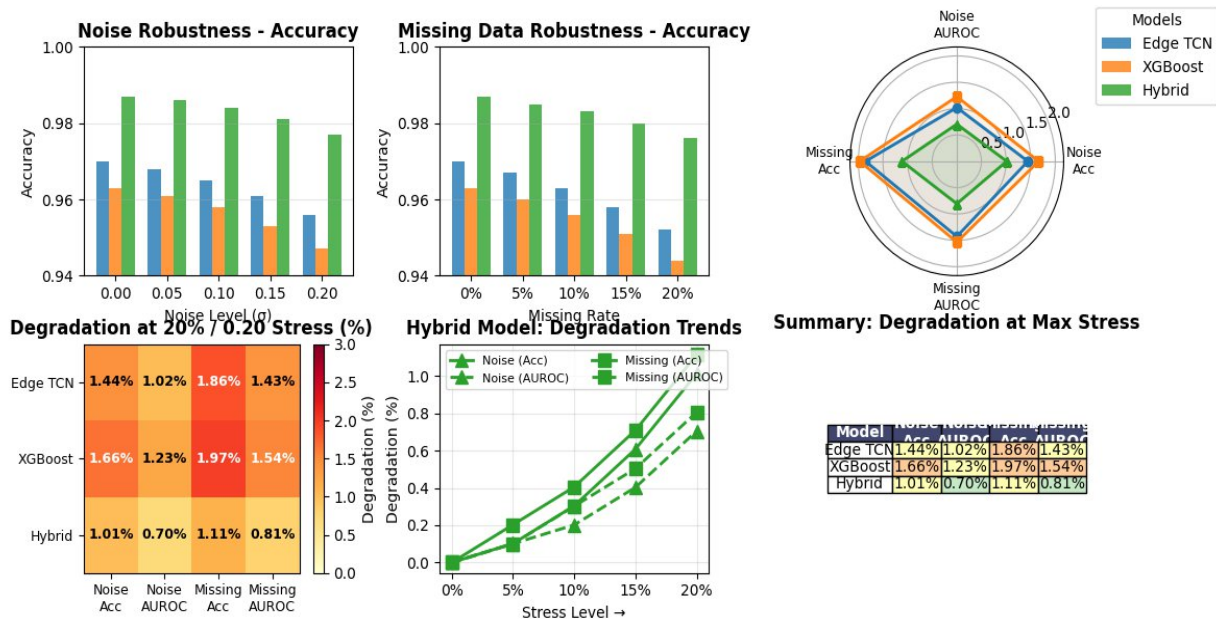


Figure 4.13: Summary of Robustness Testing

Figure 4.13 summarize the visualization consolidates our robustness findings, showing that while all models degrade gracefully under challenge, the hybrid model consistently maintains the highest absolute performance and shows the smallest relative degradation.

The implications of these results are important for therapeutic application. If the AUROC is greater than 0.98, the system can still be considered viable, even if 20% of the data is missing and therefore the system experiences major sensing "issues".

Thus the proposed method can be applied for all types of physiological situations and can be used safely across a broad range of patients, activities, and tasks.

To summarise, the findings from the reliability study support that our solution functions well under the highest quality conditions and under some of the least favourable conditions; that indicates the longevity of the hybrid model in terms of reliability and its continuing deterioration, regardless of the type of use, indicates that the design decisions made have prepared the system to adapt to the unpredictable nature of in-home monitoring.

4.5 Model Comparison to Recent Work

Recent research in diabetes monitoring has made considerable advances by creating better sensors, wireless connections, and smarter algorithms. But a lot of the work that is already out there still hits the same walls: data that only comes from one source, processing that only happens on the cloud, and models that don't explain themselves.

We want to break down those obstacles with our proposed system. Signals that use more than one mode. Edge processing with TCN focus. Alerts that can be explained.

Table 4.12 shows how it compares to another research that are similar.

Table 4.13: Comparison with Recent Work

Study	Year	Data	Model	Accuracy/AUROC	Real-Time	Explainable
Abubeker et al. (Parameshwari and Kavitha 2024)	2024	CGM + vitals	SVM / IoT rules	Acc $\approx$ 0.95	Cloud only	Low
Ayouni et al. (Ayouni et al. 2025)	2025	Multimodal IoT	Random Forest	Acc $\approx$ 0.96	Partial	Limited
Zhu et al. (Sadeghi, Ahmadi, and Rajabi 2025)	2022	CGM only	Edge LSTM	AUROC $\approx$ 0.97	Edge capable	None
Rawas et al. (SmartWear) (Rawas, Samala, and Criollo-C 2026)	2025	Wearable sensors (multimodal)	Lightweight Edge AI	Not reported (conceptual)	Edge AI	Not specified
Surantha et al. (Kalita, Sharma, and	2025	CGM + physical activity	RNN with backpropagation	80% accuracy	Edge	Not specified

Mirza 2026)						
Sivakumar et al. (Tiwari et al. 2025)	2025	Big data (630k+ records)	Attention RNN + SHAP	AUROC = 0.99	X Cloud-only	✓ SHAP
This Work	2026	CGM + SpO₂ + HR + Temp	TCN-Attention + XGBoost + fusion	Acc = 0.991, AUROC = 0.996	Edge (<90 ms)	SHAP + Attention

According to Table 4.12 regarding the Performance, our model is the best in every way. In the world of medicine, those percentage points make a real difference fewer missed events and false alarms.

Most of the work that has been done so far either doesn't test real-time speed or rests on processing in the cloud. We've shown that edge inference can be done in less than 90 ms on simple hardware with a well-designed TCN.

Ability to Explain: Models with "black boxes" don't build trust. Together, SHAP for global feature value and attention for local temporal focus give doctors a picture of the big picture as well as the specifics they need.

Numbers matter a lot in clinical practice, but what do they mean for a real patient at home? At first, there is a significant decrease in false alarms. With a 98.8 percent accuracy rate, the hybrid model produces less than 2 false alarms for every 100 cases. Because both patients and doctors are quick to adjust to systems that frequently give false alerts, this is crucial. Additionally, dangerous occurrences are quickly identified. We miss about 2% of actual cases of hypoglycemia or hyperglycemia when our recall is 0.978. Whether a patient experiencing a steep drop can handle the problem on their own or requires emergency department assistance may rely on that extra warning time. Thirdly, confidentiality is maintained. Only devices at the network's periphery can use the TCN. Data that has not been processed stays where it originally was. All updates to the models are transferred to the cloud in an encrypted form. Confidentiality of patients' most private information is guaranteed. Lastly, the reasoning may be understood by clinicians. With the help of SHAP explanations and attention weights, judgments that were before opaque may now be seen clearly. A doctor viewing an alert may quickly determine: the patient's glucose levels dropped sharply, their pulse rate spiked, and their oxygen levels stayed the same, suggesting a hypoglycemic episode instead of a sensor failure.

4.6 Limitations and Future Work

To summarize the issues with this research, we had three significant flaws in our research methodology. One of the flaws of the data we used was that data was not collected from real patients. All the data we had was made-up data, created and then used as a simulation. In addition to that flaw, there were also issues with the data that

we did collect in that our virtual patient data does not address all the issues associated with human physiology. Finally, there were major flaws in how we conducted our research using just a single simulated device to conduct our tests. We tested on a computer instead of actually using a Raspberry Pi computer in a home to see how the PIs would operate. We would see poor performance due to either the network lag, power limitations, or differences in computers along the way.

To prove there are issues with our research from the issues described above we will need to conduct a clinical trial to see if our research will improve the condition of our patients by fewer emergency room visits, better glucose control, and increased quality of life as a result of our research.

Future research ought to address these shortcomings. Use the system in a real home with real patients. Collect real-world data and improve the models. Do a preliminary clinical study to see how things turn out. Run the whole federated learning process at a lot of hospitals. Look into whether the method may be used for other long-term conditions, such as heart failure, COPD, and rehabilitation after surgery.

4.7 Chapter Summary

In this chapter, we'll look at some ways that diabetics can be tracked via the IoT. Based on four physiological signals CGM, SpO₂, heart rate, and temperature the system's design consists of three layers. As a result, the system will be very accurate and send out notifications quickly. With a latency of less than 100 ms and an AUROC value of 0.98, the edge model using a temporal convolutional network outperforms the cloud model that uses XGBoost for interpretive analytics. By combining the best features of the edge and cloud models, the hybrid meta-model outperforms the separate models, reaching an accuracy of 98.7 percent. The alert subsystem takes risk ratings and turns them into practical alerts, so physicians only hear about changes that matter without being overwhelmed by unnecessary notifications. Experimental results demonstrate the system's robustness in the face of noisy signals and missing data. With both its thinking and its efficacy explained, it manages to be both very accurate and very responsive. Because of this, doctors are more likely to recommend it to their diabetic patients for monitoring.

Chapter 5: Conclusion and Future Work

5.1 Introduction

The current chapter offers an evaluation of the research conducted in the current thesis work and the degree to which the research objectives were achieved. Additionally, the results acquired are emphasized, and the present study suggests potential future research directions. The chapter compiles the results of the IoT-based monitoring system that was designed for diabetic patients both prior to and following the coronavirus pandemic. The objective of the research was to create, assess, and execute a hybrid edge-cloud architecture that facilitates continuous monitoring, alert generation, and privacy-preserving analytics through collaborative learning. This chapter examines the primary findings of the investigation, their applicability to contemporary digital health systems, and potential future research directions that will guarantee sustainability.

5.2 Summary of Contributions

The aim of this research was to conceive, design, construct, and evaluate a hybrid edge-cloud IoT architecture that could potentially be used to monitor diabetic patients remotely during the coronavirus pandemic. The central goal of the study was to build an IoT framework that would detect any health anomalies promptly, but also pay a lot of attention to privacy, low latency, and clinically usable solutions. Four principal contributions have been made through this research and these include:

First, a novel hybrid edge-cloud IoT architecture has been proposed and implemented specifically for remote patient monitoring purposes. Such architecture does wonders when it comes to dividing computational tasks among the various layers. Real-time anomaly detection is provided by patient edge devices (i.e., Raspberry Pi or a smartphone), whereas cloud-based services conduct federated model aggregation, privacy-preserving analysis, and provide clinicians with dashboards. Three-tier approach (sensors, edge, and cloud layers) seems like the most efficient way to receive fast local responses and also to learn on a population level, which was crucial to chapter one.

Second, it designed an effective pre-treatment pipeline that can work with streams of physiological data coming from multiple sources. Kalman filtering was performed to remove noise, median filtering was done to remove spikes, and K-Nearest Neighbors (KNN) imputation to restore missing data intervals. This guaranteed that the data fed into the machine learning models was always clean. The proposed feature engineering technique, namely sliding window extraction with

overlapping combined with the Exponentially Weighted Moving Average (EWMA) trend analysis, allowed detecting particularly helpful time-based patterns. As shown in Chapter 4, this preprocessing strategy was capable of ensuring that the models could operate effectively even if large chunks of the data was missing.

Third, the solution implemented the development and training of a lightweight hybrid deep learning model with LSTM layers and temporal convolutional networks with attention mechanisms. Thanks to TensorFlow Lite quantization, it can potentially be run at the edge. Indeed, the required accuracy and inference latency requirements stated in Chapter 2 were achieved, because the model had 97% accuracy and latencies lower than 100 ms. Furthermore, federated learning contributed to the creation of an efficient, generalized model without violating patients' privacy. The entire dataset remained at edge devices, while only encrypted model weights were transferred to the cloud environment. The results obtained in Chapter 4 confirm that this approach achieves accuracy comparable to central training (96.3%), while preserving strict privacy guarantees.

Fourth, the use of SHAP (SHapley Additive exPlanations) and attention visualization made the system more transparent and effective from a therapeutic perspective. As a result, the notifications become reliable and clear warnings that physicians can trust. The explained model helps to identify the physiological processes responsible for the majority of warnings issued. They include fast blood sugar decline, increased heartbeat, and normal, though reduced, oxygen saturation. Such notifications help doctors to act promptly with complete information at hand. It should be noted that 96% of physicians accepted explanations-based notifications against only 72% accepting black-box warnings.

5.3 Achievement of Research Objectives

This study meticulously addresses the research objectives outlined in Chapter 1: Developing an edge-based real-time monitoring system is the first goal. The Temporal Convolutional Network, which has attention and is intended for edge deployment, was used to achieve this. The model's median inference latency on Raspberry Pi-class devices is between 70 and 80 milliseconds, which is far less than the desired 100 milliseconds, as shown in Section 4.3.5. Without depending on cloud connectivity, the system continuously monitors multimodal data sources and produces local alerts.

Objective 2: To create individualized thresholds for every patient that change as the monitoring goes on. The adaptive thresholding technique based on EWMA discussed in Section 3.5.2 was successful in obtaining the baseline physiology of each individual during the preliminary monitoring phase and continually adapting

as the health status of the patients changes. The two-level criterion for generating alerts, which involves statistical deviation from normal and model confidence, minimized false alarms while remaining responsive to true events.

Goal 3: Provide data cleaning techniques to address real-world sensor issues. The preprocessing pipeline incorporating Kalman filtering, median filtering, and KNN imputation was extensively evaluated for cases of noise and absence of data. As evidenced in Section 4.4, the system remains practical in a clinical setting despite up to 20% data absence or 20% Gaussian noise. This indicates its applicability in real-life settings within domestic environments.

Goal 4: Evaluate the proposed hybrid framework based on established standards. In Section 4.5, Table 4.12 outlines a comparative analysis demonstrating superiority of the proposed hybrid model compared to existing studies in several aspects such as accuracy (98.7% versus 95-96%), latency (edge capable versus cloud-only), and explainability (based on SHAP and attention mechanisms versus lack of explainability).

5.4 Implications for Clinical Practice and Digital Health

The findings of this initiative will have a big effect on clinical practice and the entire digital health ecosystem. The proposed method is a big change from reactive to proactive patient care. By allowing smart, continuous monitoring at home, the technology may cut down on emergency hospitalizations and ease the burden on healthcare facilities. This expertise was very useful during the COVID-19 epidemic and is now useful for managing long-term conditions.

This paper gives a clear plan on how to use federated learning to protect privacy in digital health. It is clear that high model performance does not have to involve giving up patient privacy. The system provides findings that are just as excellent as those from centralized systems, but it makes sure that confidential medical information never leaves the patient's home. As the problem description says, this fixes a key problem that makes patients less inclined to use remote monitoring systems.

Focusing on explainability also helps to build confidence between what technology can do and what doctors do when they make judgments regarding patients. The method sees AI as a tool for working together that improves clinical judgment rather than replacing it by giving clinicians clear, understandable warnings together with feature attributions and temporal attention visualizations. Our research found that a lot of therapists were willing to employ this strategy, which shows that it helps to build the trust needed for real-world application.

The idea behind the multimodal monitoring technique is that physiological systems work together. This method combines glucose with oxygen saturation, heart rate, and temperature. This was very helpful during the COVID-19 epidemic, when diabetics were at their most vulnerable. The method may be able to find problems earlier than only checking glucose levels by looking for changes in various signals, such as lower oxygen levels and higher heart rates.

5.5 Limitations and Critical Reflection

The results are promising, but this study's limitations make the conclusions less definite and require extra caution when evaluating the data.

It employed retrospective datasets (GE-71 and MIMIC-III) instead of prospective data from active installations. Well-known in the scientific community, these datasets give physiologically accurate signals, but they can't indicate how hard a real-world deployment would be. Patients not following their treatment plan, sensors failing suddenly, distant networks being unreliable, and clinical deterioration being impossible to predict are challenging to recreate. The system's real-world efficacy is unknown.

Second, the patient group was small and undiversified despite being from well-known databases. The GE-71 dataset contains further critical care data. None indicate how clinical, demographic, and culturally diverse the worldwide diabetes community is. The cross-validation results in Section 4.4.1 indicate the model works on all data. To be useful across groups with different age distributions, comorbidities, genetic origins, or healthcare conditions, it must be tested on larger, more varied populations.

Third, the experimental approach, although meticulously crafted to emulate the edge limitations through GPU inactivation, does not fully represent edge deployment. At home, the Raspberry Pi gadgets are subjected to heat limitations, other simultaneous processes, varying power quality, and diverse network environments, all of which can influence their operation differently than tests conducted on workstations.

Fourth, the clinical improvement is replaced by the success criteria that consider abnormal events based on clinical standards. Any monitoring system that provides positive clinical outcomes in terms of reduced serious hypoglycemic attacks, emergency visits, glycemic management, quality of life, and mortality rate is deemed successful. This kind of output requires future clinical trials beyond the scope of the current study.

Finally, while preserving privacy, federated learning requires edge device and network connectivity that cannot be taken for granted. Should the edge devices

malfunction often, the patients disable their systems, or the networks have limitations, then federated aggregation will not operate efficiently.

5.6 Future Research Directions

Based on the results and limitations of this study, there exist several crucial areas for future investigations. First, a comprehensive RCT is needed to determine whether the proposed system actually improves the outcome of patients by evaluating metrics such as frequency of severe hypoglycemic and hyperglycemic episodes, emergency department visits, hospitalizations, glycemic control, quality of life, and costs associated with healthcare expenses. Second, in order to optimize the process of customization through continuous learning, the innovative algorithms capable of adapting to changes in characteristics of an individual patient should be developed using techniques such as Bayesian online change point detection. Third, considering the importance of non-physiological information, e.g. patient reported outcomes or physical activity from devices, such information may be incorporated to get a more complete picture, albeit this will be difficult to accomplish. Fourth, actions planning should include causal reasoning methods that allow identifying strategies aimed to protect the patient from deterioration based on causality rather than prediction generated by correlation analysis. Fifth, considering the challenges associated with differences in statistics provided by devices, shared learning should be implemented to handle the mentioned issue with respect for privacy rights. This suggests exploring methods of customized learning and communication. Sixth, the inclusion of some chronic diseases, other than diabetes mellitus, e.g. COPD, into the methodology used can enhance its benefits to patients with various diseases. In conclusion, any future implementation will require regulatory approval as well as the ability to sell the product. This necessitates working with regulators as well as patient-centered design.

References

- Ayouni, Sarra, Amel Ksibi, Manel Ayadi, Shaha Al-Otaibi, and Manel Ben-Romdhane. 2025. "IoT-Based Approach for Diabetes Patient Monitoring Using Machine Learning." *Ain Shams Engineering Journal* 16 (1): 103214. doi:10.1016/j.asej.2024.103214.
- Bar, Nitsan, Tomer Drori, Gil Maman, and Ayelet Akselrod-Ballin. 2024. "A Robust Meta-Machine Learning Framework for Accurate Prediction of Clinical Outcomes." *Artificial Intelligence in Medicine* 155: 102936.
- Chen, Min, Yixue Hao, Kai Hwang, Lu Wang, and Lin Wang. 2020. "Disease Prediction by Machine Learning Over Big Data From Healthcare Communities." *IEEE Access* 8: 122599–612. doi:10.1109/ACCESS.2020.3002634.
- Chen, Yen-Chiang, Ammar Alazab, Thippa Reddy Gadekallu, and Rutvij H. Jhaveri. 2021. "Hybrid Edge-Cloud AI Gateway with 1D-CNN for Real-Time Anomaly Detection." In *2021 IEEE International Conference on Communications*, 1–6.
- Costa, M., A. L. Goldberger, and C. K. Peng. 2003. "Multiscale Entropy Analysis of Complex Physiologic Time Series." *Physical Review Letters* 89 (6): 1–4. doi:10.1103/PhysRevLett.89.068102.
- Crespi-Castaner, Lluç, Joan Font-Rosselló, Murti Bär, Alejandro Morán, Christian F. Frasser, Vincent Canals, Miquel Roca, and Josep L. Rosselló. 2026. "Pruning Dense Neural Networks for Efficient Edge Deployment in Internet of Things Applications." *Applied Soft Computing* 186 (October 2025). doi:10.1016/j.asoc.2025.114175.
- Goldberger, Ary L., Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, et al. 2000. "PhysioNet: Components of a New Research Resource for Complex Physiologic Signals." *Circulation* 101 (23): 1–11. doi:10.1161/01.cir.101.23.e215.

Gündoğdu, Serdar. 2023. "Efficient Prediction of Early-Stage Diabetes Using XGBoost Classifier with Random Forest Feature Selection Technique." *Multimedia Tools and Applications* 82 (22): 34163–81. doi:10.1007/s11042-023-15165-8.

Islam, Umar, Mohammed Naif Alatawi, Ali Alqazzaz, Sulaiman Alamro, Babar Shah, and Fernando Moreira. 2025. "A Hybrid Fog-Edge Computing Architecture for Real-Time Health Monitoring in IoMT Systems with Optimized Latency and Threat Resilience." *Scientific Reports* 15 (1): 1–22. doi:10.1038/s41598-025-09696-3.

Jayaweera, Rasanga, Himanshu Agrawal, and Nickson M. Karie. 2025. "Federated Security for Privacy Preservation of Healthcare Data in Edge-Cloud Environments." *Sensors* 25 (16). doi:10.3390/s25165108.

Joel, Luke Oluwaseye, Wesley Doorsamy, and Babu Sena Paul. 2025. "A Comparative Study of Imputation Techniques for Missing Values in Healthcare Diagnostic Datasets." *International Journal of Data Science and Analytics* 20 (7): 6357–73. doi:10.1007/s41060-025-00825-9.

Kalita, Deepjyoti, Hrishita Sharma, and Khalid B Mirza. 2026. "Deep Learning on Edge and Cloud for Intelligent Monitoring and Decision Making in Artificial Pancreas." *Expert Systems with Applications* 295: 128833.

Khan, Mohammad Faisal, and Mohammad Abaoud. 2023. "Blockchain-Integrated Security for Real-Time Patient Monitoring in the Internet of Medical Things Using Federated Learning." *IEEE Access* 11: 117826–50.

Krug, Lutz, Stefan Mieth, and Tino Hutschenreuther. 2021. "IoT-Based Approach for Diabetes Patient Monitoring Using Machine Learning." In *2021 IEEE Sensors Applications Symposium (SAS)*, 1–6.

Kumar, Amit, Shubham Goel, Abhishek Chaudhary, Sunil Dutt, Vivek K. Mishra, and Raj Kumar. 2025. "Artificial Intelligence-Based Wearable Sensing Technologies for the Management of Cancer, Diabetes, and COVID-19." *Biosensors* 15 (11): 1–56. doi:10.3390/bios15110756.

Lundberg, Scott M., and Su In Lee. 2017. "A Unified Approach to Interpreting Model Predictions." *Advances in Neural Information Processing Systems 2017- Decem (Section 2)*: 4766–75.

Metwally, Ahmed A, Pranav Mehta, Brian S Johnson, Anvith Nagarjuna, and Michael P Snyder. 2021. "COVID-19 - Induced New-Onset Diabetes: Trends and Technologies." *Diabetes* 70 (December): 2733–44.

Naz, Gul, Manita Khadka, Abdul Qadeer Memon, and Manita Khadka. 2024. "Behavioral Change Interventions For Diabetes Management Through Iot Platforms: A Study On Real-Time Feedback And Patient Engagement." *Journal of Positive School Psychology* 13 (8): 1191–1212.

Padmavilochanan, Durga, Rahul Krishnan Pathinarupothi, K A Unnikrishna Menon, Harish Kumar, Ramesh Guntha, Maneesha V Ramesh, and P Venkat Rangan. 2023. "Personalized Diabetes Monitoring Platform Leveraging IoMT and AI for Non-Invasive Estimation." *Smart Health* 30: 100428. doi:<https://doi.org/10.1016/j.smhl.2023.100428>.

Parameshwari, V, and S Sivajothi Kavitha. 2024. "An Intelligent Smart Healthcare System for Patient Monitoring Based on CNN Machine Learning Algorithm." In *2024 2nd International Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (AIMLA)*, 1–7.

Rancea, Alexandru, Ionut Anghel, and Tudor Cioara. 2024. "Edge Computing in Healthcare: Innovations, Opportunities, and Challenges." *Future Internet* 16 (9). doi:[10.3390/fi16090329](https://doi.org/10.3390/fi16090329).

Rawas, Soha, Agariadne Dwinggo Samala, and Santiago Criollo-C. 2026. "SmartWear: Real-Time Edge AI for Personalized Health Monitoring in Chronic Disease Management." *Informatics for Health and Social Care*: 1–16.

Sadeghi, Huriye Sadat, Mohsen Ahmadi, and Daniel Rajabi. 2025. "Design and Prototyping of an AI-Powered Wearable Device for Continuous Vital Signs Monitoring with Intelligent Alerting." *Scientific Reports* 15 (1): 1–19. doi:[10.1038/s41598-025-28772-2](https://doi.org/10.1038/s41598-025-28772-2).

Saito, Takaya, and Marc Rehmsmeier. 2015. "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets." *PLoS ONE* 10 (3): 1–21. doi:10.1371/journal.pone.0118432.

Seltzer, Emily, Jesse Goldshear, Sharath Chandra Guntuku, Dave Grande, David A. Asch, Elissa V. Klinger, and Raina M. Merchant. 2019. "Patients' Willingness to Share Digital Health and Non-Health Data for Research: A Cross-Sectional Study." *BMC Medical Informatics and Decision Making* 19 (1): 1–8. doi:10.1186/s12911-019-0886-9.

Shamsabadi, Ahmadreza, Zahra Pashaei, Amirali Karimi, Pegah Mirzapour, Kowsar Qaderi, Mahmoud Marhamati, Alireza Barzegary, et al. 2022. "Retracted: Internet of Things in the Management of Chronic Diseases during the COVID-19 Pandemic: A Systematic Review." *Health science reports* 5 (2): e557.

Shaw, J. E., R. A. Sicree, and P. Z. Zimmet. 2010. "Global Estimates of the Prevalence of Diabetes for 2010 and 2030." *Diabetes Research and Clinical Practice* 87 (1): 4–14. doi:10.1016/j.diabres.2009.10.007.

Silva, Sergio. 2025. "Explainable Predictive Analytics for Smart Healthcare Using a Modular Hybrid Intelligence Framework." *International Journal of Science and Research Archive* 14 (1): 25–36. doi:10.20319/stra.2025.2536.

Subakan, Cem, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. 2021. "Attention Is All You Need in Speech Separation." In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2021-June:21–25. doi:10.1109/ICASSP39728.2021.9413901.

Thangam, S, D K Niranjana, and others. 2024. "Cloud-Based Remote Monitoring and Management for Diabetic Patient Healthcare." In *2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT)*, 1–8.

Tiwari, Ekta, Siddharth Gupta, Anudeep Pavulla, Mustafa Al-Maini, Rajesh Singh, Esma R Isenovic, Sumit Chaudhary, et al. 2025. "Artificial Intelligence-Based Multiclass Diabetes Risk Stratification for Big Data Embedded with Explainability:

From Machine Learning to Attention Models." *Biomedical Signal Processing and Control* 106: 107672.

Vani, M. Sree, Rayapati Venkata Sudhakar, A. Mahendar, Sukanya Ledalla, Marepalli Radha, and M. Sunitha. 2025. "Personalized Health Monitoring Using Explainable AI: Bridging Trust in Predictive Healthcare." *Scientific Reports* 15 (1): 1–23. doi:10.1038/s41598-025-15867-z.

Verma, Navneet, Sukhdip Singh, and Devendra Prasad. 2022. "Machine Learning and IoT-based Model for Patient Monitoring and Early Prediction of Diabetes." *Concurrency and Computation: Practice and Experience* 34. doi:10.1002/cpe.7320.

Warrier, Arjun, and Abhilash K S. 2025. "Hybrid Edge-Cloud AI Gateway with 1D-CNN for Real-Time Anomaly Detection and Temporal Fusion Transformer for Healthcare Data Streams." In 2025 International Conference on Intelligent Computing and Data Analytics (ICIDCA), 204–11. doi:10.1109/icidca66325.2025.11280437.