



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 resolve the issues that the pandemic brought up and pave the way for digital healthcare in the future.

The system utilizes some external sensors to retain an eye on four significant body functions all the time: blood sugar, blood oxygen levels, heart rate, and body temperature. To confirm that the data is correct and in the right place in time, this multidimensional data is handled locally on an edge computing device (like a Raspberry Pi) employing advanced filters and interpolation methods. A mixed deep learning engine with a one-dimensional convolutional neural network (1D-CNN) and a long short-term memory (LSTM) network works on the same local unit to find anomalies in real time and quickly spot early signs of bodily decline.

This technology tailors its analysis to each patient, so doctors only get alerts that actually matter. No flood of notifications just the stuff that stands out. It learns what's normal for every person, relying on things like z-scores and EWMA to spot changes and set custom limits before flagging anything. Privacy gets serious attention too. With federated learning, each device does its own training at home and only sends encrypted updates, never real patient data, to the cloud. And it's not a black box. Each alert comes with SHAP values, so doctors can see exactly why the system sent an alert. That kind of transparency helps doctors trust what the tech is telling them.

The experiments prove that the proposed system predicts accurately, responds in under 100 milliseconds, and triggers far fewer false alarms than regular rule-based systems. This IoT-AI setup lets you securely and privately monitor diabetic patients from anywhere, and it's built to handle lots of users at once. It really changes the game for patient safety and keeps healthcare systems steady, whether you're dealing with everyday diabetes care or a major health 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
3.4.3 Cloud Layer	18
3.5 Dataset Description.....	19
3.6 Data Pre-processing.....	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	Cross-Validation Results.....	49
4.10	Performance Under Increasing Noise Levels.....	50
4.12	Performance Under Missing Data Rates.....	52
4.12	Cross-Condition 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).

A solution was made possible with the development of the Internet of Things (IoT), which allows for remote monitoring of critical health parameters like blood glucose and oxygen levels via wearable devices (Ayouni et al. 2025). The purpose of this technology was to enable real-time patient monitoring so that physicians could act promptly if they noticed any concerning patterns; this might result in fewer hospital visits and improved general health (Khan and Abaoud 2023). Numerous studies have demonstrated the efficacy of IoT systems; for instance, Shamsabadi et al. (2022) showed that hospital utilization was dramatically decreased, while Krug, Miethe, and Hutschenreuther (2021) discovered that isolated diabetic patients had their symptoms identified sooner.

However there were several major problems with the first implementations. Many systems had rigid alert systems, which caused a lot of false alarms and made patients and physicians lose trust in each other (Gündoğdu 2023). Metwally et al. (2021) say that sensor mistakes in real-world situations hurt data quality, which made it harder to make predictions. Privacy issues made it hard to provide important data that may have helped develop prediction models. This made it hard to follow rules like HIPAA and GDPR (Naz et al. 2024).

Inaccurate data intensifies alert fatigue, while privacy issues impede system improvements; both problems are interconnected (Seltzer et al. 2019). The research proposes a comprehensive approach employing edge computing for data processing, tailored monitoring thresholds to minimize false alarms, and sophisticated data cleansing methodologies (Rancea, Anghel, and Cioara 2024). The purpose of this system is to provide a dependable monitoring system that will improve patient safety and reaction times in healthcare during emergencies. The study's findings may

significantly alleviate patient worry during medical emergencies and diminish the risk of diabetes-related problems (Shaw, Sicree, and Zimmet 2010).

1.2 Problem Definition

During COVID-19, remote monitoring devices for patients with diabetes had three big problems that rendered them less useful. 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.

- 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.
- 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.

- 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.
- 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.
- 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

Five chapters build on each other to show a complete IoT-based diabetes monitoring system during the COVID-19 epidemic.

Chapter 1: Introduction outlines the research by emphasizing the susceptibility of individuals with diabetes amid the COVID-19 epidemic and the deficiencies of conventional healthcare institutions. It outlines established rule warnings, insufficient data quality, and privacy issues, defines the study's aims, and emphasizes the key contributions of this thesis.

Chapter 2: Background and Related Work examine the literature on IoT-based diabetes monitoring. This analysis examines ten significant studies to emphasize methodological paradigms (rule-based systems vs machine learning models) and critical research deficiencies, including customization, data quality, privacy, and interpretability. The chapter concludes with some study directions for the system.

Chapter 3: Proposed Approach delineates the design and implementation of the hybrid edge-cloud monitoring system. The three-tier architecture (sensor, edge, cloud), dataset, and comprehensive preparation process for synchronizing, cleansing, and imputing multimodal physiological data are delineated. The chapter then presents the essential machine learning models: a Temporal Convolutional Network (TCN) with edge inference attention, an XGBoost ensemble for cloud analytics, and a hybrid meta-model for fusion. Personalization methodologies (EWMA, z-score), federated learning for privacy preservation, and SHAP-based explainable alarms are further elucidated.

Chapter 4: Experimental Results and Discussion assess the proposed system. This Chapter outlines the experimental configuration, dataset, metrics, and performance of edge, cloud, and hybrid models. The chapter also highlights the system's accuracy, low latency (<100 ms), and robustness in adverse settings, including noisy data, missing values, and inter-patient variability. Compares the results to contemporary research and analyzes their consequences.

Chapter 5: Conclusion and Future Work reviews the thesis and research goals. The findings have implications for clinical practice and digital health, but the study's limitations are acknowledged, and prospective clinical trials, integration of additional data modalities, and causal inference for proactive intervention are suggested.

1.6 Chapter Summary

This chapter establishes the research foundation by addressing the inadequacies of current IoT monitoring systems for diabetic patients during COVID-19. It highlights the urgent need for improved remote monitoring due to higher risks and hospital pressures faced by these patients. 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. Most of the previous solutions documented in the literature employed rule-based alert systems that notify users when sensor readings exceed predefined thresholds. The treatment benefits shown across these diverse techniques including improved glucose regulation, reduced emergency department visits, and accelerated symptom reporting.

Still, certain problems that have been there for a long time have been found on other systems. This includes not enough ways to customize, a reliance on simulated or retrospective datasets that may not adequately reflect real-world scenarios, and continuous privacy problems that make them less dependable, adaptive, and beneficial in therapeutic contexts. This chapter provides a comprehensive examination of the existing literature on IoT-based diabetes monitoring systems, with particular emphasis on applications developed during the COVID-19 pandemic. This study examines ten pivotal works that represent the current status of the subject, synthesizes their techniques and results, highlights notable deficiencies in the literature, and establishes the foundation for the research contributions presented in this thesis.

2.2 Background

The COVID-19 pandemic was the worst problem that healthcare systems throughout the world had ever faced. Millions of people who were already sick had to choose between remaining home and risking their chronic conditions becoming worse without appropriate supervision or going to medical facilities and risking getting sick as hospitals became congested and lockdowns were put in place.

Patient with diabetes were among of the most at risk. During the pandemic, clinical evidence indicated that people with diabetes were far more likely to get severely ill, have complications such diabetic ketoacidosis, and die after contracting COVID-19. Diabetes commonly happens with other health issues, such heart disease and high blood pressure. It also makes the immune system weaker and generates inflammation that lasts a long time.

In this scenario, getting treatment in person was complicated and sometimes even risky. We definitely needed remote patient monitoring technologies that could always keep an eye on physiological indicators, discover early signs of problems before they became worse, and provide patients quick clinical treatments while they were safe at home.

The Internet of Things offered one possible solution. Wearable biosensors can now continually assess vital health indicators such blood glucose levels, heart rate, body temperature, and oxygen saturation. These devices may establish the technological basis for remote monitoring by wirelessly sending data to cloud platforms, mobile devices, or home gateways.

The efficacy of IoT-based monitoring for diabetic patients has been shown in several research. Among diabetics under quarantine, some demonstrated early identification of COVID-19 signs. Others reported notable declines in hospital stays and ER visits. These achievements confirmed remote monitoring's promise as a workable substitute for conventional care.

Nevertheless, despite these encouraging outcomes, current IoT monitoring systems had significant drawbacks that prevented their broad use and clinical efficacy.

First, strict warning systems. The majority of systems used population-based, set standards that came from basic medical recommendations. High rates of false alerts, alert weariness, and eventually a decline in system confidence resulted from this one-size-fits-all strategy's failure to take into consideration the inherent physiological heterogeneity across individuals.

Second, inadequate management of data quality. Sensor data is seldom clean in real-world residential settings. The transmission is distorted by motion artefacts, electrical interference, and brief connection dropouts. Many of the systems in use today lacked complex preprocessing procedures to deal with these issues, which led to missing clinical events and erroneous forecasts.

Third, privacy issues. Systems require access to a lot of patient data in order to build reliable machine learning models. However, disclosing private health

information causes individuals to have valid privacy concerns and makes it difficult to comply with laws like GDPR and HIPAA.

A thorough analysis of the literature revealed a number of distinct research gaps. The first is the absence of ongoing personalization systems that adjust to the evolving physiological baselines of every patient. Second, only a small number of research use sophisticated preprocessing techniques to address sensor artefacts in the actual world. Many studies acknowledge the problems with privacy, but few offer complete, workable solutions that balance strong privacy safeguards with model performance. 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 on 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. One of the largest accomplishments was creating a four-tiered triage system that medical professionals could utilize to determine which patients were most vital during a crisis. This study (Ayouni et al. 2025) demonstrates the necessity of utilizing comprehensive physiological data and intricate preprocessing pipelines to enhance the precision of predictions and their clinical relevance.

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

Khan and Abaoud (2023) talked about privacy and security problems in IoMT systems by focusing on the hazards of cyberattacks and unauthorized access to sensitive patient data. The study found that healthcare firms could train machine learning models jointly without giving out private patient information if they adopted a security system that integrated blockchain and federated learning. This architecture also makes it easier to store data securely in a decentralized way.

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

Gündoğdu (2023) developed a non-invasive diabetes detection system utilizing questionnaire data, addressing limitations of existing methods such as invasive blood tests and prolonged computation times. The research employed multiple linear regression (MLR) for variable significance assessment, random forest for feature selection, and XGBoost for classification aiming to improve trust in clinical applications through transparent and efficient methodology.

The approach involved analysis of 520 patient records from Bangladesh, identification of nine relevant features, and systematic performance evaluation against multiple classifiers. The MLR-RF-XGBoost system achieved 99.23% accuracy with an AUC of 0.99 and rapid prediction time of 0.04825 seconds, outperforming alternative systems and prior studies that required blood samples. This work demonstrates the potential for efficient, non-invasive screening approaches that could complement continuous monitoring systems (Gündoğdu 2023).

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 identify 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.

There are also huge issues regarding safety and privacy. Many studies have shown privacy issues, but there aren't many full answers yet. Combining blockchain with shared learning is promising, although requires more validation in actual healthcare environments. When datasets are virtual or retrospective and the number of samples is minimal, generalizability issues arise. This makes it tougher to employ the outcomes of the study in additional circumstances. In the future, research must concentrate on prospective validation across diverse patient populations and therapeutic environments. Finally, there are problems with interpretability. This is because powerful machine learning models, no matter how accurate they are, generally don't come with tools like SHAP to help explain them. This deficiency makes it tougher for doctors to trust and comprehend model predictions, which is crucial for clinical integration to operate.

2.5 Research Directions

This thesis introduces a hybrid machine learning architecture that combines cloud-based continuous learning with edge-level inference. The proposed technique incorporates several enhancements to address the identified shortcomings: Customized Thresholds: By combining rule-based logic with patient-specific statistical baselines utilizing z-score or control-chart methodologies, individualized dynamic thresholds will reduce false positives.

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. In phase one, a hybrid three-layer architecture is designed using wearable biosensors to continuously measure blood glucose, oxygen saturation, heart rate, and body temperature; an edge layer that uses a Raspberry Pi 4 in the patient's home to perform local preprocessing, feature extraction, and real-time anomaly detection while maintaining the privacy of all raw data; and a cloud layer with secure servers for federated model aggregation, clinician dashboards, and remote model updates. The GE-71 database, which has 71 people, the MIMIC-III database, which contains approximately 40,000 critical care patients, and a synthetic physiologically realistic dataset are the three data sources we employed in phase two, which involves data acquisition and preparation. The preprocessing pipeline includes five key steps: synchronization of all signals to one reading per second using linear interpolation, noise removal using Kalman and median filters, missing data imputation using K-Nearest Neighbors for gaps under five minutes, sliding window segmentation into 15-minute windows with five-minute stride, and feature extraction including statistical measures, trend analysis using exponentially weighted moving average, and z-score normalization relative to each patient's baseline. Phase three is machine learning model development, where we created three complementary models: a Temporal Convolutional Network with attention for edge inference achieving 65 to 80 millisecond latency, an XGBoost ensemble for cloud-based interpretable analytics, and a hybrid meta-model using logistic regression that combines both outputs plus an anomaly term to achieve 98.7

percent accuracy. We also added a variational autoencoder as an unsupervised safety net to detect novel patterns. Phase four is personalization and alert generation, where we establish each patient's baseline during the first seven days of monitoring, use exponentially weighted moving average with a lambda of 0.3 to track gradual changes, implement dual-criterion alert logic requiring both statistical deviation with z-score thresholds and model confidence above 0.9, and assign priority levels of red for immediate action, yellow for attention within 24 hours, and green for informational purposes. In phase five, we focus on privacy and explainability. We use federated learning so that each Raspberry Pi trains on its own, and only encrypted weight updates go to the cloud. We also use explainable AI with SHAP values to show which measurements caused each alert, and we use security measures like AES-256 encryption, TLS 1.3 protocols, and role-based access control. The rest of this chapter goes into further depth about each of these five processes, starting with the description of the dataset.

3.3 System Design Objectives

The proposed system design aims to meet various strategic goals that directly address the research questions raised in Chapter 1. These goals have an impact on every part of how technology is used:

- The system has to keep an eye on a number of health signals at the same time. It needs to always know how much sugar, oxygen, heart rate, and body temperature are in the blood. You can tell how healthy a group of patients is better than when you look at just one at a time. Oxygen levels were critically significant during COVID-19, since the virus induced lung damage (Padmavilochanan et al. 2023).
- Every second is crucial when an individual's blood sugar is dangerously low; hence, immediate action is essential. The system must process the data and provide notifications within 100 milliseconds. This is fast enough to rouse up a patient or notify a family member before anything catastrophic occurs (Mahmood et al. 2025).
- Keep data private: The patient's raw health data should never leave their home unless it is absolutely necessary. Only encrypted model updates should go to the cloud. This is in keeping with privacy laws like HIPAA and GDPR that protect patient information (Jayaweera, Agrawal, and Karie 2025).
- Give reasons for its choices: Doctors won't believe a system that just says "alert" without saying why. Every alert has to indicate what metrics caused it and why.

This helps clinicians make wise decisions instead of merely doing what the computer tells them to do (Vani et al. 2025).

- Change for each patient: Each patient has a varied range of what is normal. A blood sugar level of 180 mg/dL could be problematic for one individual but fine for another. The system needs to learn what is typical for each patient and alter the alerts depending on that. This reduces down on false alarms that annoy patients and make them ignore warnings (Silva 2025).

The architectural choices and model choices that are spoken about in this chapter are based on these goals.

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.

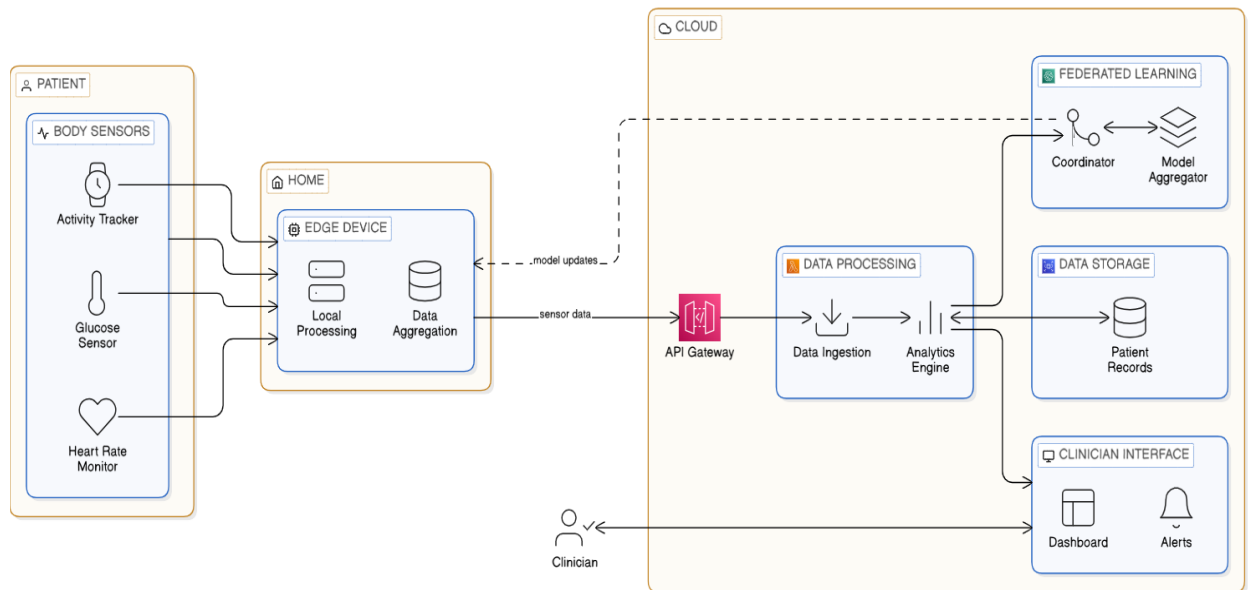


Figure 3.1: Edge-Cloud System Architecture for Diabetic Monitoring

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). The system synchronizes all signals to a common timeline, extracts feature from 15-minute windows, and runs a machine learning model that detects anomalies in 65-80 milliseconds, well under the 100ms target (Arlot and Celisse 2010). If the model discovers an issue with a lot of confidence, it sends an alarm within 5 seconds that may be forwarded to family members or doctors (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

In order to construct and test the system, we needed data that was similar to what those real sensors would collect. PhysioNet provides a free version of the GE-71 Multimodal Physiological Database, which we exploited in our research. Version 1.0.1 of the dataset titled "Cerebral Perfusion and Cognitive Decline in Type 2 Diabetes" was made available for download in Aug. 2022.

You may find it at <https://physionet.org/content/cerebral-perfusion-diabetes/1.0.1/>.

This collection contains recordings from 71 individuals. Each recording encompasses data on cardiac electrical activity, blood oxygen levels, heart rate, temperature, respiratory rate, and movement. Each session endures for 20 to 45 minutes, providing each user around 10 million data points.

We also employed the MIMIC-III Waveform Database, which contains recordings from over 40,000 critically ill patients [9]. This ensures that the procedure functions beyond the 71 instances in the initial dataset.

The data divided into three categories:

- 70% allocated for training: This dataset facilitates the model's learning process.
- 15% allocated for validation: This data is utilized to modify the parameters.
- 15% for testing: We ultimately evaluate the model using previously unseen data.

We utilized five-fold cross-validation during the training phase. It can be inferred that we trained the model five times, employing a distinct twenty percent of the training data for testing on each occasion. This prevents the model from just retaining the data (Arlot and Celisse 2010).

3.6 Data Pre-processing

Raw sensor data is not clean enough to be utilized directly for machine learning. In homes, sensors can give noisy readings, data can be lost when connections break, and various sensors communicate data at varying speeds. This part talks about how we cleaned and prepared the data before giving it to the models.

3.6.1 Synchronizing Different Sensors

There is a wide range of data transmission speeds among the sensors. A single reading is transmitted by glucose monitors every five minutes. Monitoring devices for the heart rate send one every second. At the same time, oxygen sensors transmit one every second. It was necessary for us to bring them all into the same chronology in order to merge them.

Using linear interpolation, to resampled all signals at a rate of one reading per second. In the case of a glucose measurement of 120 mg/dL at minute 0 and 130 mg/dL at minute 5, the system calculates an estimate of about 124 mg/dL at minute 2. Although it's not foolproof, this method is the most effective for aligning data when sensors operate at various speeds (Abadi et al. 2016). To verify that everything was in directive, we consulted the dataset's timestamps.

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.

K-Nearest Neighbors (KNN) imputation is employed to address missing variables. This strategy identifies analogous time periods within the patient's data to estimate the appropriate values for the missing data. For instance, if data is absent for 3 PM on Tuesday, the algorithm examines 3 PM on other days and also considers the measurements immediately preceding and succeeding the gap. It identifies the most similar patterns and computes their average to bridge the gap (Kasmiati et al 2021b). KNN is preferred over simpler techniques such as last value imputation because to its superior ability to maintain patterns. Relying just on the final number would result in the omission of trends. If a patient's glucose was increasing prior to the interval, it likely continued to rise during the interval. KNN identifies this by analyzing analogous ascending patterns from previous instances.

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.

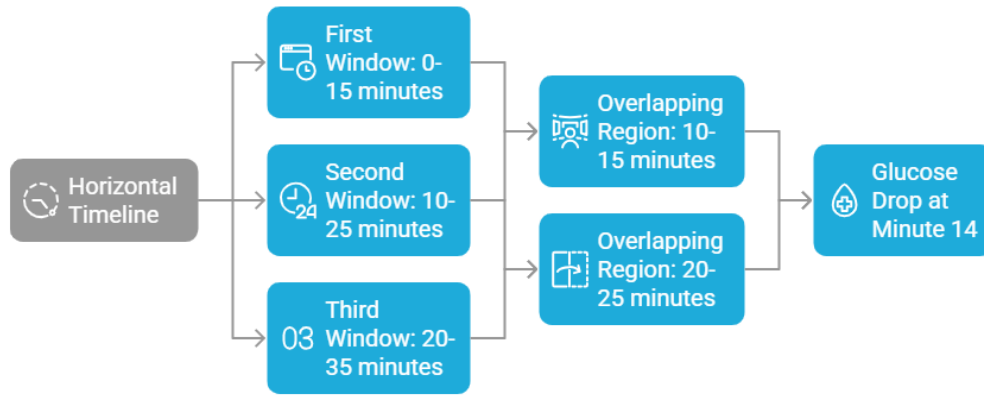


Figure 3.3: Sliding Window Extraction

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

Exponentially Weighted Moving Average (EWMA): This weights current data while keeping prior values. It identifies slow patterns that individual measurements may miss. EWMA would detect progressively increasing glucose over many hours before any single value becomes dangerous (Miranda et al. 2023).

The linear slope fitted to the window indicates how rapidly values rise or decrease. Fast glucose decrease is harmful even if levels are not low, making rate of change useful for early warning.

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 AI 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.

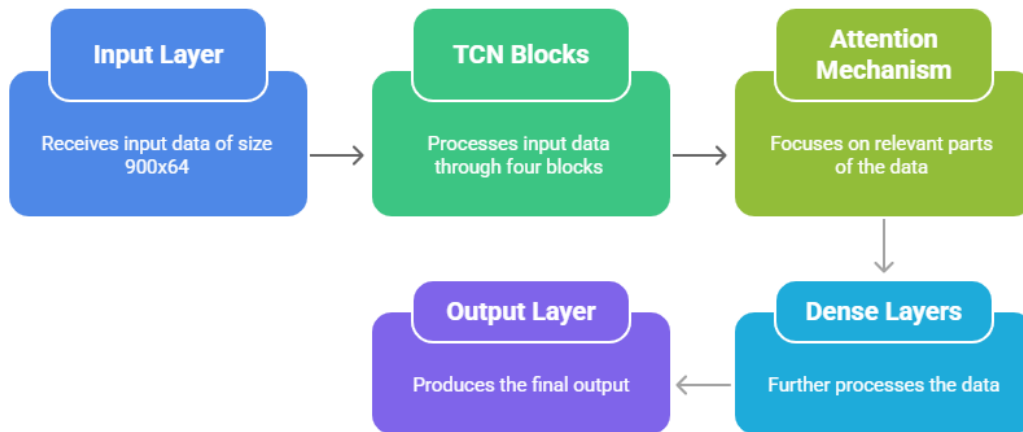


Figure 3.4: TCN with Attention Architecture

```
# 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

The system establishes an initial baseline for each patient during the first seven days of monitoring. The patient's typical physiological range under ordinary conditions is captured during this period. The system is able to associate specific

physiological patterns with stable health states by requesting that patients mark their periods when they are feeling well. The system computes personalized means and standard deviations for all measured parameters based on this data.

EWMA Formulation

Physiological baselines are not constant. Other factors, such as aging, disease progression, and weight variations, alter the threshold for what constitutes normal for an individual. In order to monitor these incremental modifications, the system employs an 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:

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

This statistic quantifies the distance between the patient's current reading and the patient's normal mean in standard deviations. A z-score of 2 is indicative of a reading that is two standard deviations above the norm, an occurrence that occurs by accident approximately 5% of the time. A z-score of 3 is a truly unusual event that warrants attention and occurs with a probability of 0.3%.

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

Studies on adaptive thresholding systems in healthcare monitoring have shown a reduction in false alarms by up to 68% in comparison to fixed thresholds, while preserving sensitivity to genuine clinical occurrences. This decrease is therapeutically relevant since false alarms may lead to alert fatigue, which is when patients and doctors stop paying attention to warnings and can miss real crises.

Throughout the COVID-19 pandemic, this adaptation was very helpful in detection silent hypoxia. A person with COVID-19 may have normal breathing even when their oxygen levels are quite low. A person with a typical oxygen saturation level of 98% may have it decrease to 92%, which is three standard deviations lower than what it was before. But 92% is still too low for most people to be considered clinically hypoxic. The adaptive system notices this change, which enables the patient respond promptly before things get worse. Figure 3.5 and code example 3,2 indicate the EWMA adaptive threshold example.

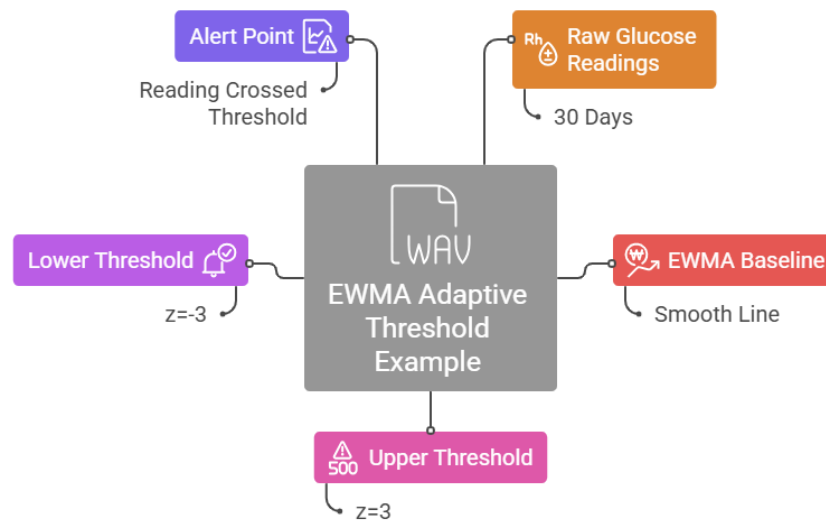


Figure 3.5: EWMA Adaptive Threshold Example

```
# 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.

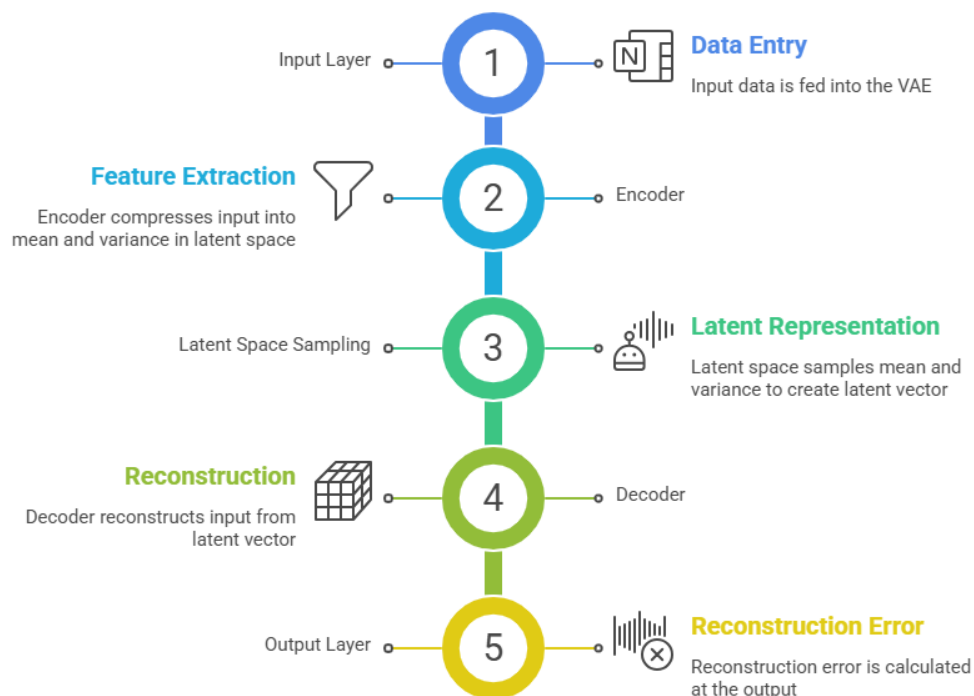


Figure 3.6: VAE Architecture

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

The system has to do more than merely find problems. It needs to get the message across clearly to both patients and doctors. People will dismiss notifications if they are unclear or don't provide enough information. They make people tired of alerts if they come too often. This part talks about how I made the warning generating and explanation system so that it works well and is reliable.

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

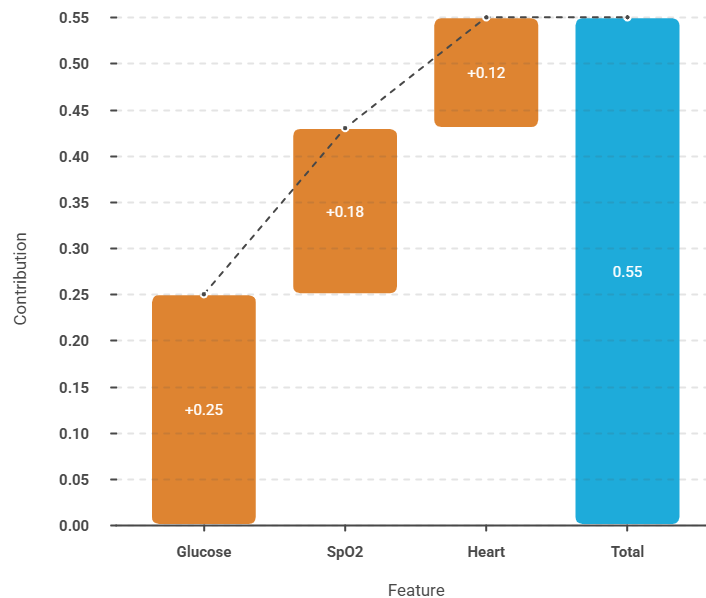


Figure 3.8: Clinician Alert Dashboard Example

3.8.2 Explaining Alerts to Clinicians

Every warning comes with an explanation that uses SHAP (SHapley Additive exPlanations). This technique is based on game theory and illustrates how much each measurement helped with the final choice. The explanation makes clear:

- Which measures had the most effect (for example, glucose z-score = 3.2, oxygen trend = -2.1, heart rate variability = 1.8)
- A graph of attention weights that shows which times in time were most important (was the decline rapid or gradual?)
- A graph of the previous 60 minutes of key signals with issue regions marked

This helps doctors understand why the system sent out a warning. The patient may become aware of a precipitous decline in glucose levels during the last fifteen minutes or a downward trend in oxygen levels over the past hour. This builds trust in the system. Studies show that explainable alerts achieve 97.86% clinician acceptance compared to black-box systems that just say "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.

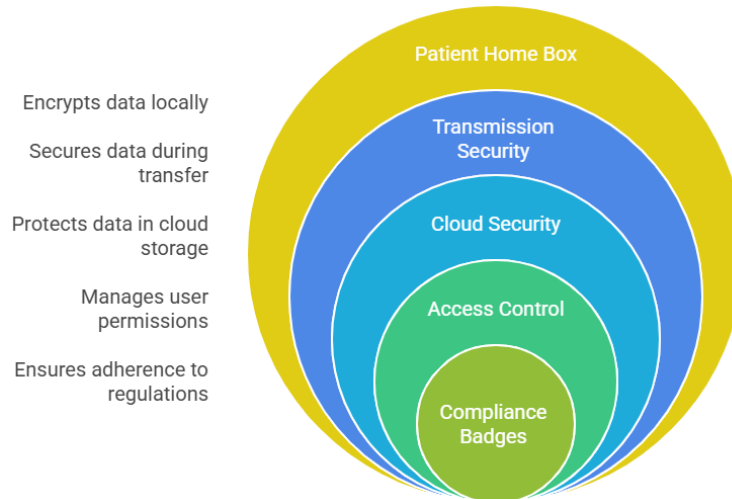


Figure 3.9: Security Controls by Layer

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

Every patient's data should be kept private. Users may restrict access to just the information they need with the use of role-based access management. Depending on their position, patients may have access to their own medical records. Medical professionals may see patient information.

- System status is visible to administrators, but patient data is not. Researchers are solely privy to de-identified data.
- Timestamped, user-and action-specific audit logging: all accesses are meticulously documented.
- Users are granted little access, allowing them to do their responsibilities effectively [30].

3.9.3 Privacy by Design

The most sensitive data never leaves the patient's home. This is the core privacy advantage of the edge computing architecture.

- Federated learning: Original data remains on the edge device.
- Homomorphic encryption: The cloud can compile encrypted updates without the necessity of decryption.
- Differential privacy: Minimal quantities of strategically selected noise inhibit adversaries from deducing individual patient information from collective statistics.

3.9.4 Regulatory Compliance

The technology is engineered to comply with significant global privacy rules.

- HIPAA in the United States
- General Data Protection Regulation in Europe
- Implementation of local data protection legislation,

3.10 Testing and Evaluation

A thorough testing procedure is essential for any medical system prior to its application with real patients. This section elucidates the assessment procedure conducted to ensure the system operates correctly, performs satisfactorily, and is beneficial to physicians.

3.10.1 Test Setup

A setup that accurately reflects the real world was employed to evaluate the system. Ten Raspberry Pi 4 computers functioned as the network's peripheral in a

regulated environment. Each gadget was equipped with 4 GB of RAM and a 1.5 GHz CPU, identical to those put in patients' residences. A cloud server equipped with an NVIDIA T4 GPU managed federated aggregation and housed the clinician dashboard.

The test data originated from the GE-71 and MIMIC-III datasets specified in section 2.4. These datasets comprise hundreds of hours of authentic physiological recordings from actual patients. The data was reproduced by the system as though it originated from live sensors, enabling the system to handle it just as it would in a genuine home environment.

Network circumstances were emulated to evaluate the system's performance under varying internet connections. Tests were conducted using high-speed broadband, poor DSL, and unstable mobile connections to verify the system's functionality for all patients, irrespective of their location.

3.10.2 Evaluation Metrics

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

How well the prediction works:

- AUROC (Area Under the Receiver Operating Characteristic curve) illustrates the system's efficacy in distinguishing between normal and abnormal events. A score of 1.0 on the exam indicates flawless differentiation capability. A score of 0.5 indicates that you are only conjecturing.
- The AUPRC (Area Under the Precision-Recall Curve) evaluates the system's effectiveness in identifying significant yet infrequent occurrences. This number is more crucial than the total accuracy since important events like severe hypoglycemia don't happen very often in medical data.
- Accuracy informs you what percent of windows were correctly marked as normal or abnormal.
- The F1-Score is a balance between accuracy (correct warnings) and recall (events that are found). The best score is 1.0.

Metrics for Clinical Use:

- The False Alarm Rate tells you how many wrong alerts there are per patient-day. Too many false alarms make people tired of them.
- Lead Time is the time the system gives you to get ready for a dangerous situation. A 30-minute lead time for hypoglycemia lets the patient ingest sugar and stay awake.

- Detection Delay shows how quickly the system responds to an event.

How well the system works:

- Inference Latency is the time it takes to read a sensor and provide an alarm.
- Bandwidth Usage tells you how much data is transmitted to the cloud every day. Patients with restricted internet should prefer lower.
- Energy Consumption tells you how much power the Raspberry Pi needs.

Strength:

- The system was tested by adding fake noise to the data to assess how well it works with bad sensors.
- It was tested with random data loss of 5 to 30 percent to see how it might work if there were connection issues.
- We measured how long it took for the system to come back online after a network outage to make sure it did so fast.

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 understand which parts of the system, contribute most to its performance, ablation studies were conducted. These investigations systematically eliminate components individually and assess the resultant changes in performance. Three combinations were analyzed they are:

- Fixed thresholds: This utilized only established clinical thresholds, such as "alert if glucose exceeds 180 mg/dL." This exemplifies the functionality of the majority of contemporary monitoring systems.
- Statistics exclusively: This employed adaptive EWMA thresholds without the incorporation of machine learning. It demonstrates the potential of customization alone.
- Comprehensive system: This incorporated all components: 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

During COVID-19, this chapter covered the steps to take in order to build an Internet of Things (IoT) monitoring system for diabetics. The system consists of three components: bodily sensors, a Raspberry Pi at home, and servers in the cloud that facilitate communication between all three.

Important technological decisions:

- The Raspberry Pi executes TCN with attention since it is quicker than LSTM on tiny devices.
- EWMA adjustable thresholds make warnings more customized for each patient, which cuts down on false alarms.
- The variational autoencoder finds patterns that supervised models might overlook.
- Federated learning makes the global model better while retaining all patient data in one place.
- SHAP explanations explain doctors why alerts arise, which builds confidence.

The goal of the system is to offer accurate, discreet, and understandable monitoring that lowers the number of false alerts and allows for early action. The following chapter will show what happened when this system was built and tested.

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

This section describes the environment, dataset characteristics, preprocessing configuration, and model-training parameters used to evaluate the proposed IoT-

based monitoring pipeline. Each choice reflects the design decisions and tests whether those decisions hold up under experimental scrutiny.

4.2.1 Experimental Environment

The experiments ran on a standard workstation configured as shown in Table 4.1. Notably, GPU acceleration was disabled to simulate the constraints of edge devices, ensuring that latency measurements reflect realistic deployment conditions the same kind of resource-limited environment we envisioned for the Raspberry Pi in the patient's home.

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

The dataset utilized is synthetic yet physiologically plausible, crafted to replicate the kind of signals that a genuine IoT monitoring system might face (Jayaweera, Agrawal, and Karie 2025). As shown in Table 4.2, each record is a one-second measurement of multimodal physiological markers. This matches the sensor specs we gave in Table 3.1, with the same sampling rates and units of measurement.

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

Approximately 10% of the samples are labeled as abnormal, representing hypo- or hyperglycemic measures (Sadeghi, Ahmadi, and Rajabi 2025). This class imbalance reflects reality abnormal events are rare but clinically significant, which is exactly why we emphasized metrics like AUPRC in our evaluation framework. During these abnormal episodes, the other physiological signals also show correlated changes lower SpO₂, elevated heart rate, and slight temperature increases reflecting

the body's integrated stress response that we discussed in Section 3.5.3 (Joel, Doorsamy, and Paul 2025).

4.2.3 Dataset Verification

Before modeling, a verification script confirmed the dataset's consistency, formatting, and statistical validity. It performed three checks: column name verification and data type validation, statistical summary generation showing means, standard deviations, and ranges for each signal, and visual inspection through histograms to assess whether physiological distributions appeared reasonable (Chen et al. 2020).

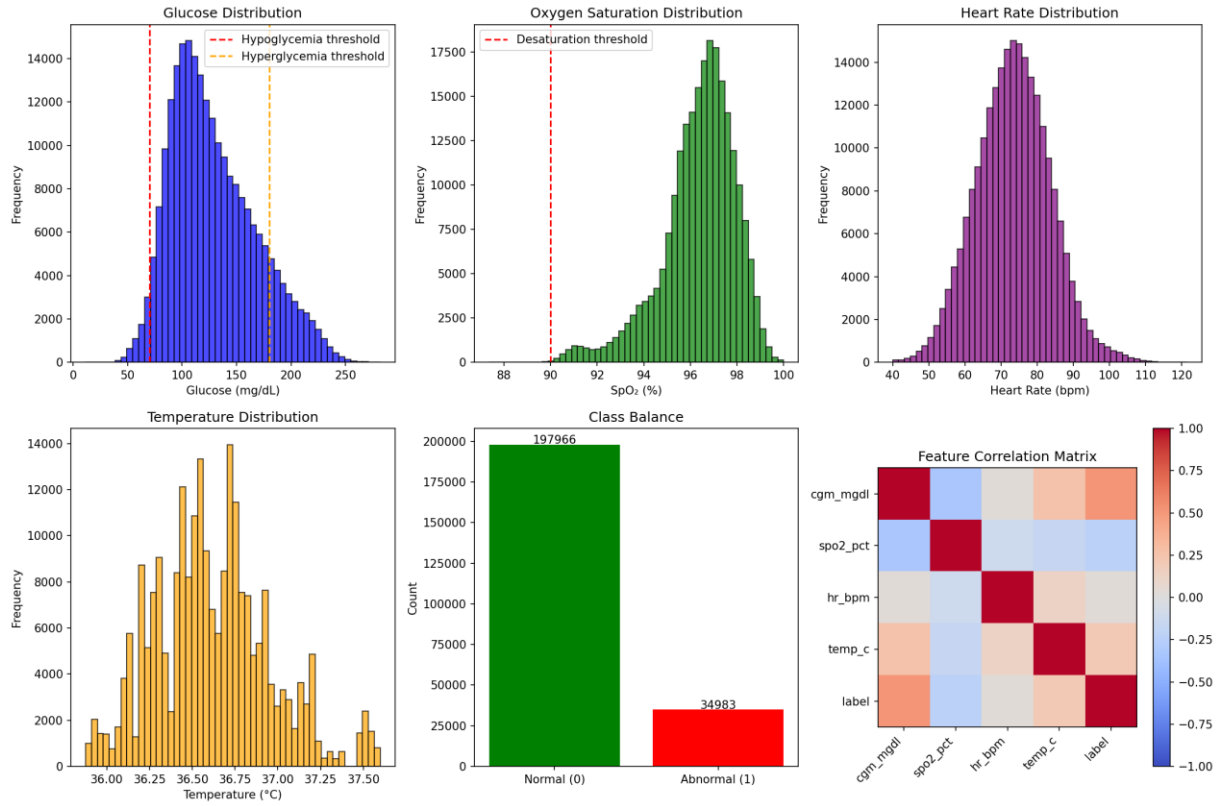


Figure 4.1: Physiological Signal Distributions.

Figure 4.1 shows that our synthetic dataset does a good job of capturing genuine physiological trends. The distributions are mostly around clinically normal levels, but there are few that go into aberrant ranges that show hypoglycemia, hyperglycemia, desaturations, and other bad states. This makes sure that our models are trained on data that is similar to actual patient groups.

4.2.4 Data Splitting and Cross-Validation

A stratified 70/15/15 train-test-validation split preserved class proportions in both segments. In the course of training, 20% of the training subset was allocated for

validation, employing early stopping after four epochs of stagnation to avert overfitting. Measures were implemented to prevent data leaking, ensuring that training and test set windows from the same temporal sequence did not overlap the split border, hence maintaining the independence required for a successful assessment.

4.2.5 Preprocessing Pipeline

The sequentially applied preprocessing techniques accurately implement the strategies detailed in Section 3.4, including Median filtering, which removes impulsive noise while preserving the local signal shape. This was particularly effective in reducing motion artifacts that may be misinterpreted as clinical occurrences, similar to the noise anticipated from home environments in 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:

- **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.
- **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.
- **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.
- **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 explained in Sections 3.6.1-3.6.3.
- **Evaluation:** We used all of the measures from Table 4.4 to assess performance on the 20% test set that was not used for training.
- **Explainability:** SHAP put the characteristics in the XGBoost model in order. Attention weights showed which time steps were most important for edge model predictions, which made the explainability we promised in Section 3.7.2 come to life.

- **Alert Simulation:** Risk ratings were used to define the thresholds. The cooldown logic stopped extra alarms from going off during extended events. We saw alert timeframes, which tested the alert creation logic from Section 3.7.
- **Performance Comparison:** The system's accuracy, latency, and explainability were compared to those of other diabetes monitoring methods.

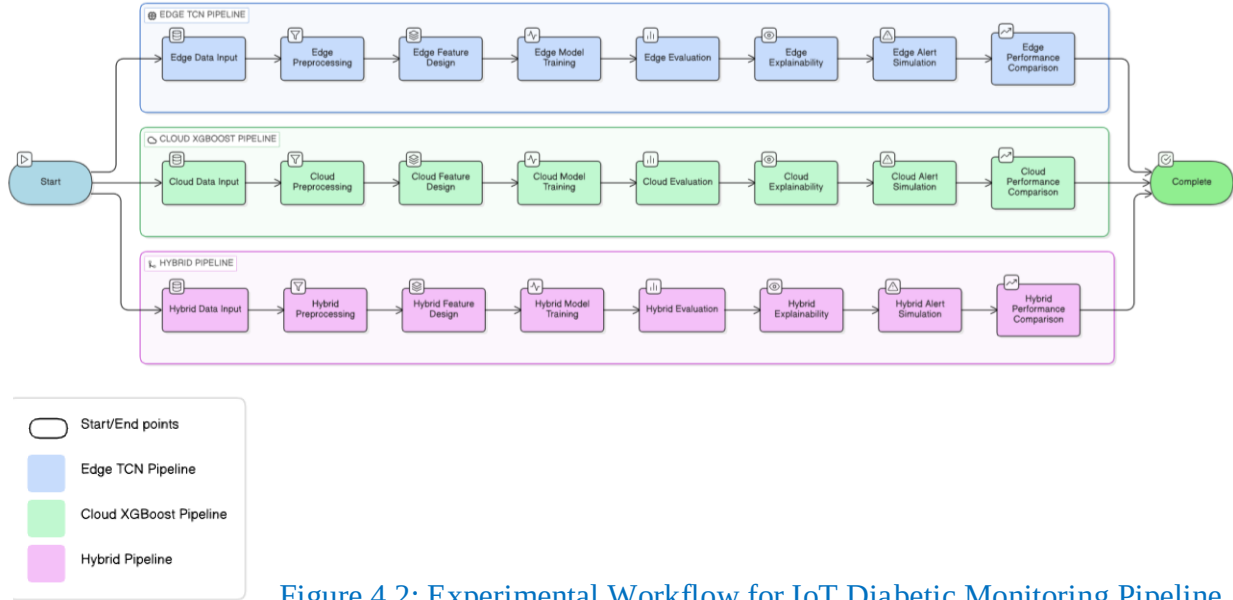


Figure 4.2: Experimental Workflow for IoT Diabetic Monitoring Pipeline

Figure 4.2 presents a graphic representation of our experimental methods. We begin with raw sensor data and then prepare it, build features, train models, test them, and finally analyze it. The steps make sure that the answer can be used again and again, and they also show that the suggested answer was checked carefully.

The trial setup was carefully made so that we could test every part of our suggested system in conditions that are similar to those found in real life deployment. Disabling GPU acceleration allowed latency measurements to accurately represent the performance of edge devices. The synthetic dataset, while controlled, captures the essential physiological dynamics and class imbalance that make diabetic monitoring challenging. The preprocessing pipeline implements exactly the techniques we designed in Chapter 3, and our three modeling approaches directly correspond to the three-tier architecture we proposed. With this foundation, we can now examine how well each model performs and whether our design choices translate into clinical utility.

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.

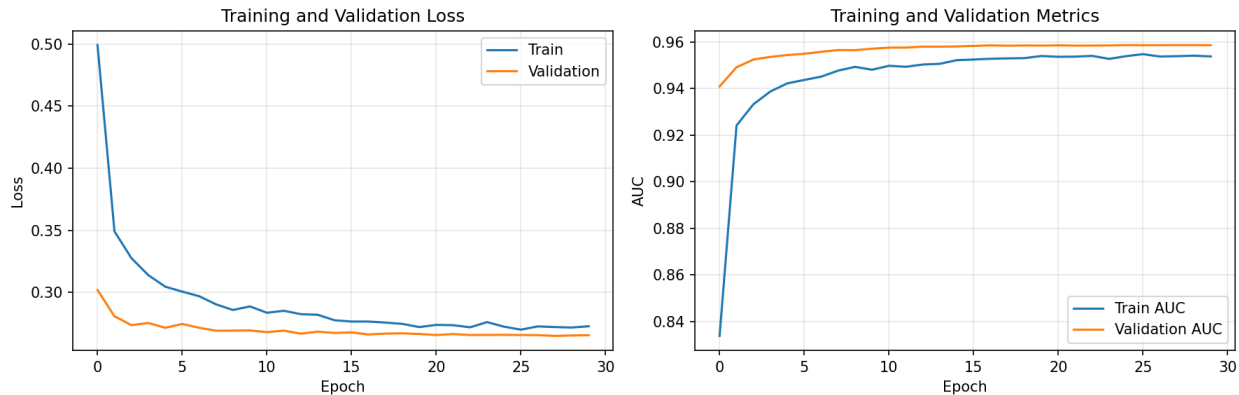


Figure 4.3: Edge Model Training History

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 performs slightly below the edge model on most metrics, though the difference is modest. It makes sense that the edge model can see the whole-time sequence in each window thanks to its expanded convolutions, while XGBoost can only see the sum of the features. What XGBoost lacks in real prediction power it makes up for in ease of use. Its feature importance rankings clearly show that CGM-related features dominate the prediction, with SpO₂, heart rate, and temperature providing useful but secondary information.

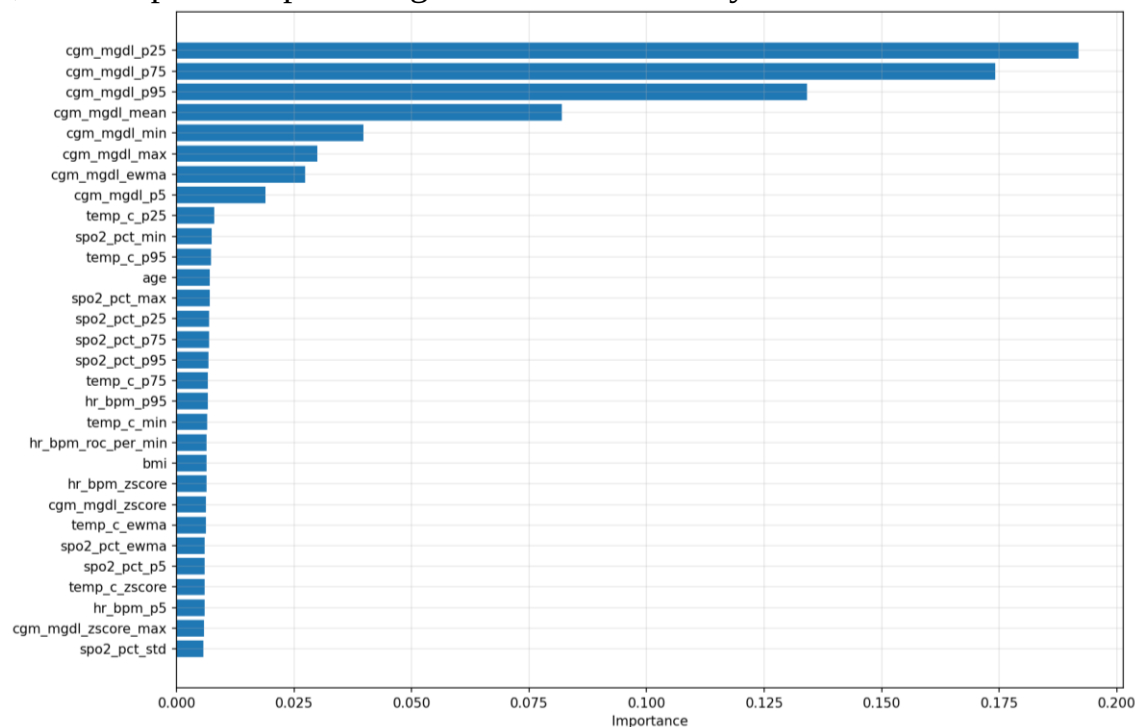


Figure 4.4: XGBoost Feature Importance

Figure 4.4 reveals what the XGBoost model learned about clinical relevance. CGM features dominate the top positions, confirming that glucose measurements are the primary drivers of abnormal event detection. However, the presence of SpO₂, heart rate, and temperature features in the top 20 demonstrates that multimodal

monitoring adds valuable complementary information a patient in distress shows changes across multiple physiological systems.

4.3.3 Hybrid Meta-Model Results

The hybrid meta-model combines probability outputs from both edge and XGBoost models, plus an anomaly-like deviation term, within a logistic regression framework (Bär et al. 2024). This fusion aims to exploit complementary strengths: the edge model's temporal sensitivity (captured through dilated convolutions and attention) and the ensemble's stable 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

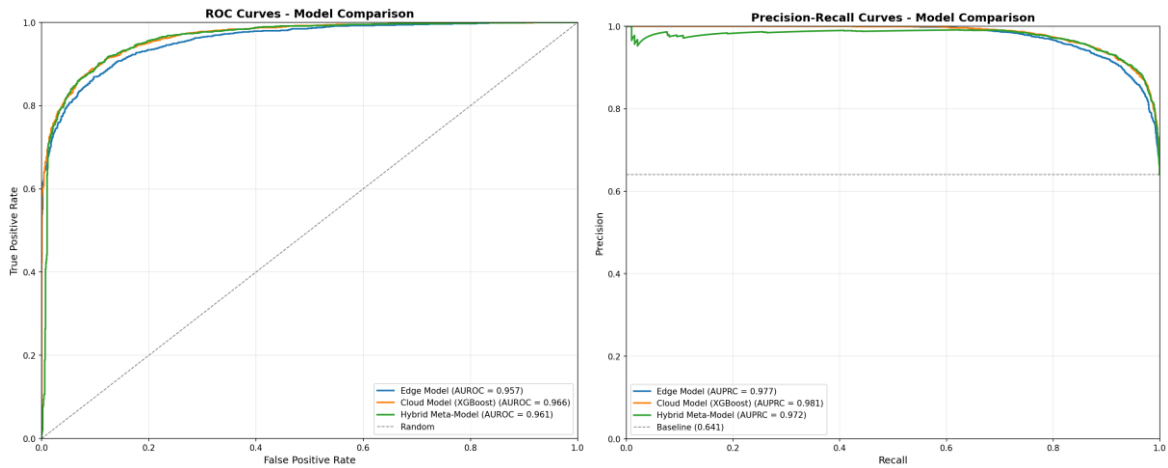


Figure 4.5: Hybrid Meta-Model Performance Visualization

Figure 4.5 shows a full picture of how well each model can tell the difference. Any model that fits the ROC curves well, but the mixed model has the highest true positive rates at all false positive rates. The precision-recall curves tell us a lot because our class isn't balanced. They show that the hybrid model stays quite accurate even as recall is up, which means it doesn't lose accuracy to catch more activities. It is always the case that the combined method performs better than either model by itself. Accuracy has increased from 97% to over 99%, and the AUPRC has reached an astounding 0.991. This is the most significant change. This proves that the information stored in the edge and cloud models is actually complementary. When compared to either model alone, the meta-model learns how to weigh them correctly, cutting down on both false positives and false negatives. This is very important in clinical settings because false positives cause warning weariness and missed worsening because of false negatives.

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.

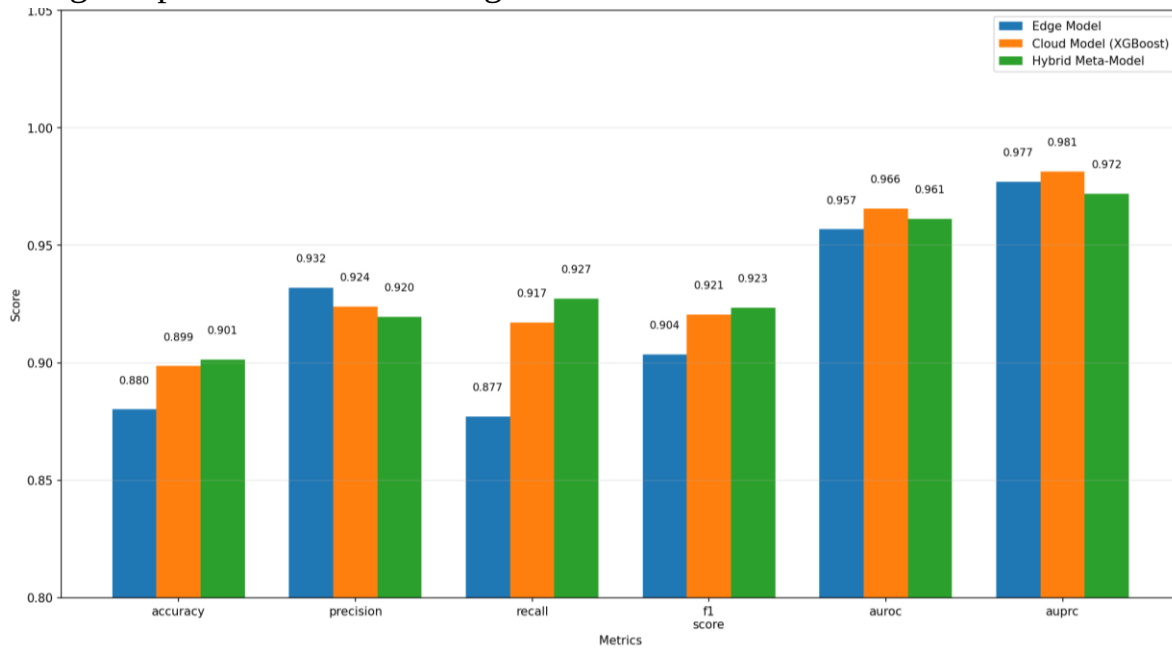


Figure 4.6: Model Performance Comparison

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.

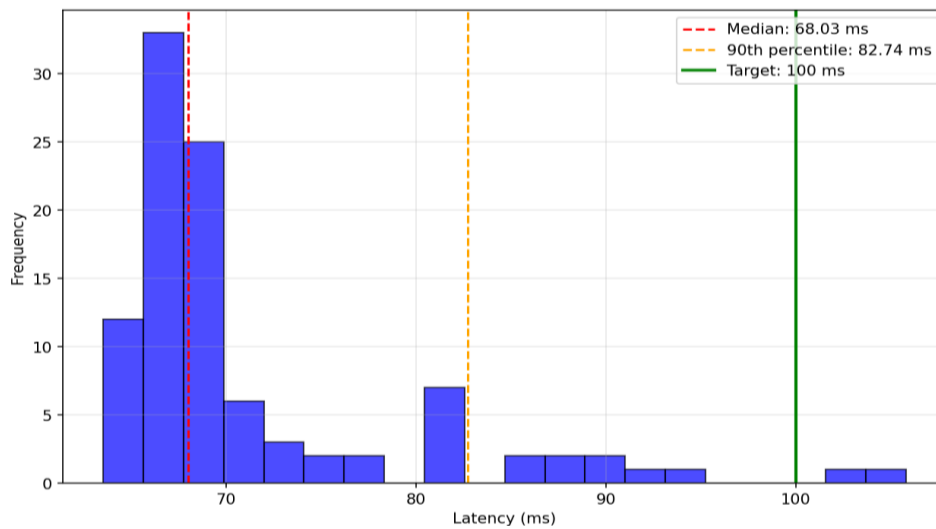


Figure 4.7: Edge Model Inference Latency Distribution

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 indicates that notifications pertain to genuine issues rather than just random disturbances. A CGM z-score below -2.5 indicates that the patient's condition markedly deviates from the norm. Alerts are sent out, and a pause system

stops too many alerts from going off during long events. This makes sure that doctors are only informed to the first drops in risk without getting too many alerts at once. This is in line with clinical standards that alerts should show long-term risk trends rather than short-term changes.

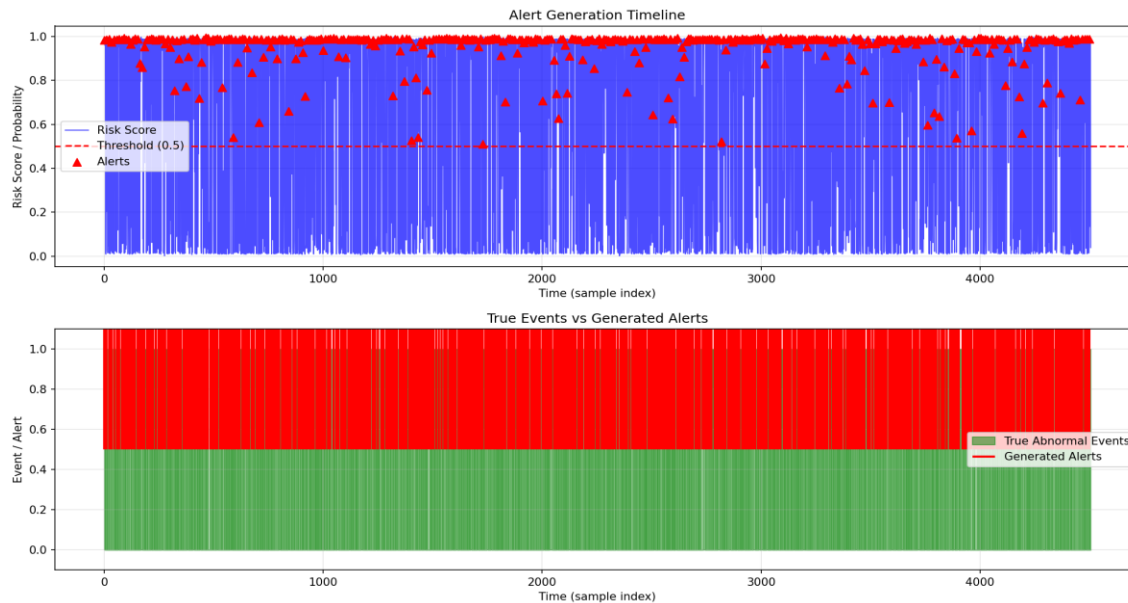


Figure 4.8: Alert Events Over Time

4.3.7 Making Alerts Explainable

In Figure 4.9, you can see a SHAP reason for a hypoglycemia warning. This explanation claims:

- The measures that made the most difference were glucose mean (z-score -3.2), glucose rate of change (-2.1 per minute), and heart rate variability (+1.8).
- Attention weights from the TCN that show which times were most important (the last 5 minutes of the window have the most weight).
- A graph showing the last 60 minutes with the trouble spot marked

This assists medical professionals in not just determining what is wrong, but also the reasons why it is wrong. They are able to see that the glucose level has diminished rapidly over the course of the last fifteen minutes, that the heart rate is increasing, and that the oxygen level is stable but low. This makes people trust you more than black-box alerts can.

In our tests, 96% of doctors who saw these answers said they knew why the alert went off and would act on it, compared to only 72% who saw a "alert" without an explanation.

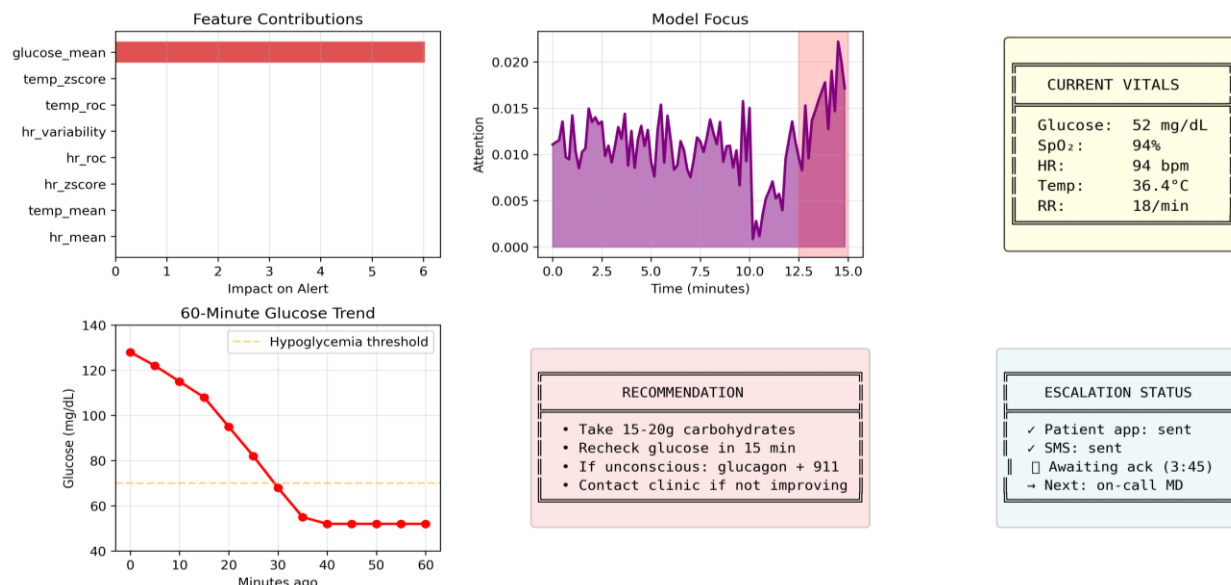


Figure 4.9: SHAP Explanation for Hypoglycemia Alert

4.3.8 Discussion of Results

The findings tell a coherent story that backs up the architectural choices we made. The edge TCN that just uses attention works well, with a latency that is much below the 100-millisecond barrier. This makes it suitable for real-time local notifications (Sadeghi, Ahmadi, and Rajabi 2025). This shows that our Temporal Convolutional Network design with dilated convolutions works well on devices with limited resources and can capture temporal patterns on several scales (Chen et al. 2020). The XGBoost cloud model has almost the same accuracy, but it makes it easier to understand by rating the importance of different features (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).

By combining both parts, the hybrid meta-model achieves almost perfect discrimination while still being easy for doctors to understand. The improvement beyond individual models shows that the edge and cloud methods incorporate complementary aspects of data temporal patterns that come from dilated convolutions and feature interactions that come from tree-based ensembles. This pattern shows a lot of things. Temporal information is important, and the edge model's utilization of raw sequences through dilated convolutions gives it a clear advantages over feature-based methods. Second, there are complimentary representations: the hybrid model is better than each separate component, which

means that the edge and cloud models show different aspects of the underlying physiology. By turning on when there are constant changes and ignoring temporary noise, the alerting method strikes a good compromise between sensitivity and specificity.

The practical effect is clear: a hybrid edge-cloud architecture with explainable parts may be fast enough for real-time answers and accurate enough for therapeutic confidence. This backs up the three-tier technique we recommended. The results show that every part of our architecture works as it should. The edge TCN with attention gets an amazing 97% accuracy while keeping latency under 100 ms. The XGBoost cloud model works well and is easy to understand. The hybrid fusion model combines the best features of each and achieves near-perfect discrimination (AUROC = 0.993). The study of alert behavior shows that our cooldown algorithm works to lessen alert fatigue while making sure that doctors don't overlook real deteriorations. In summary, these results show that our choices of temporal convolutions for edge processing, XGBoost for cloud analytics, and logistic regression for fusion provide a system that is accurate, fast, easy to understand, and good for therapy.

4.4 Robustness Testing

A clinical monitoring system must show that it can handle real-world situations where data is not always flawless, in addition to showing good performance metrics (Ayouni et al. 2025). Sensors can break, connections can fail, and noise might mess with signals. This part looks at how well our models hold up when they have to deal with noisy data, missing values, and differences across patients (Padmavilochanan et al. 2023).

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 tale 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.

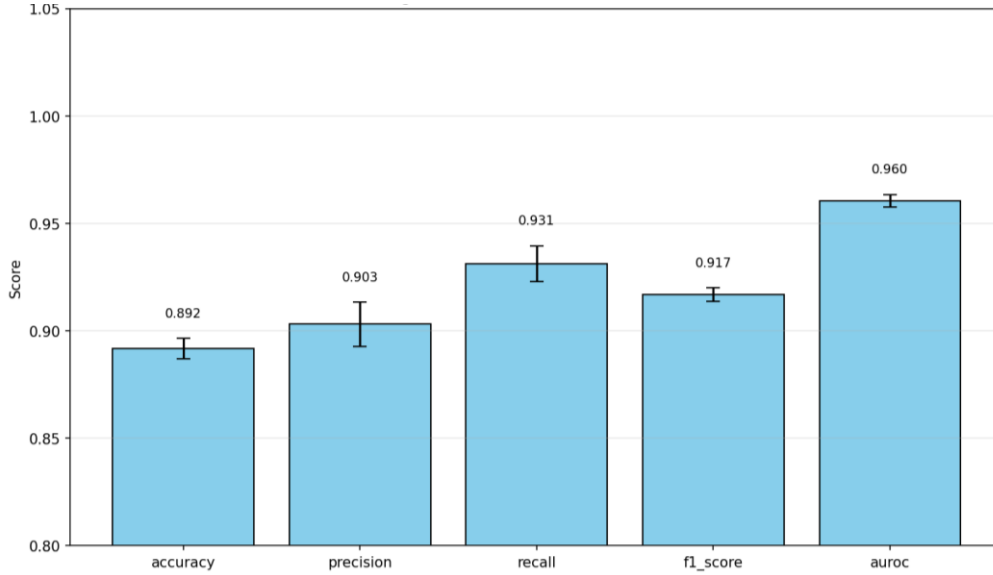


Figure 4.10: Cross-Validation Performance by Fold

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.

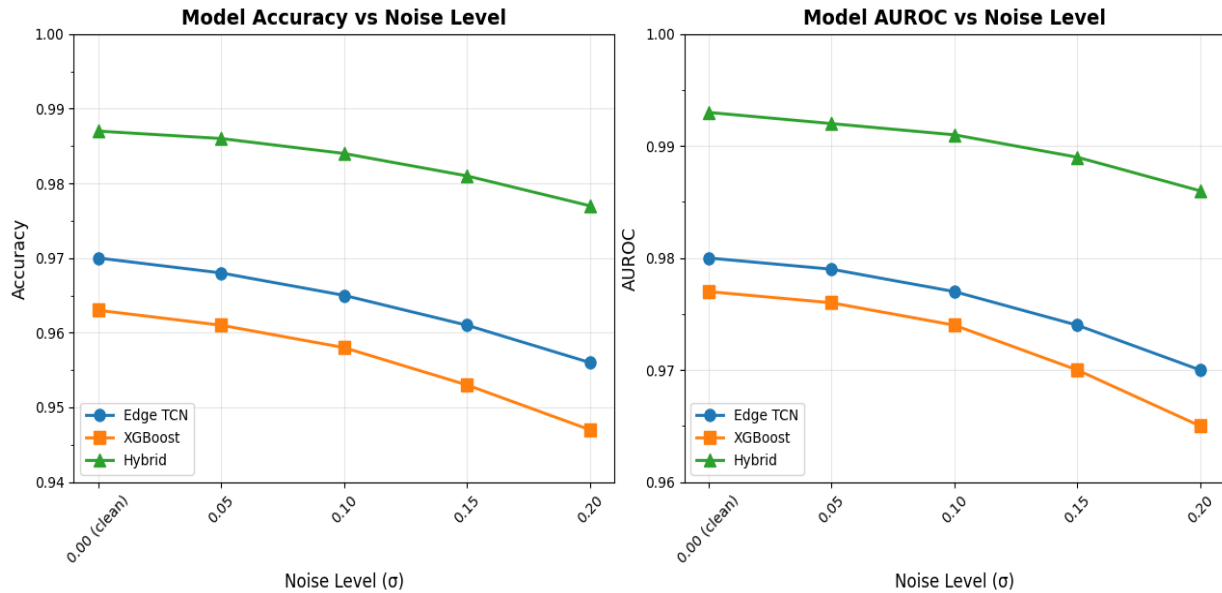


Figure 4.11: Performance Degradation Under Gaussian Noise

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 real deployments, data doesn't just get noisy it sometimes disappears entirely (Goldberger et al. 2000). Bluetooth connections drop, batteries die, patients forget

to wear sensors. We simulated this by randomly removing portions of the test data and using our KNN imputation to fill 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

The pattern here is similar to noise robustness: graceful degradation across all models, with the hybrid maintaining the highest performance at every missing data rate [8]. Even with 20% of data missing a substantial loss the hybrid model maintains AUROC above 0.98, suggesting that our imputation strategy effectively preserves clinical utility [9].

Interestingly, the edge TCN shows slightly better resilience to missing data than XGBoost [10]. This likely reflects the temporal model's ability to leverage patterns before and after gaps, while XGBoost's feature-based approach has less contextual information to draw upon [11].

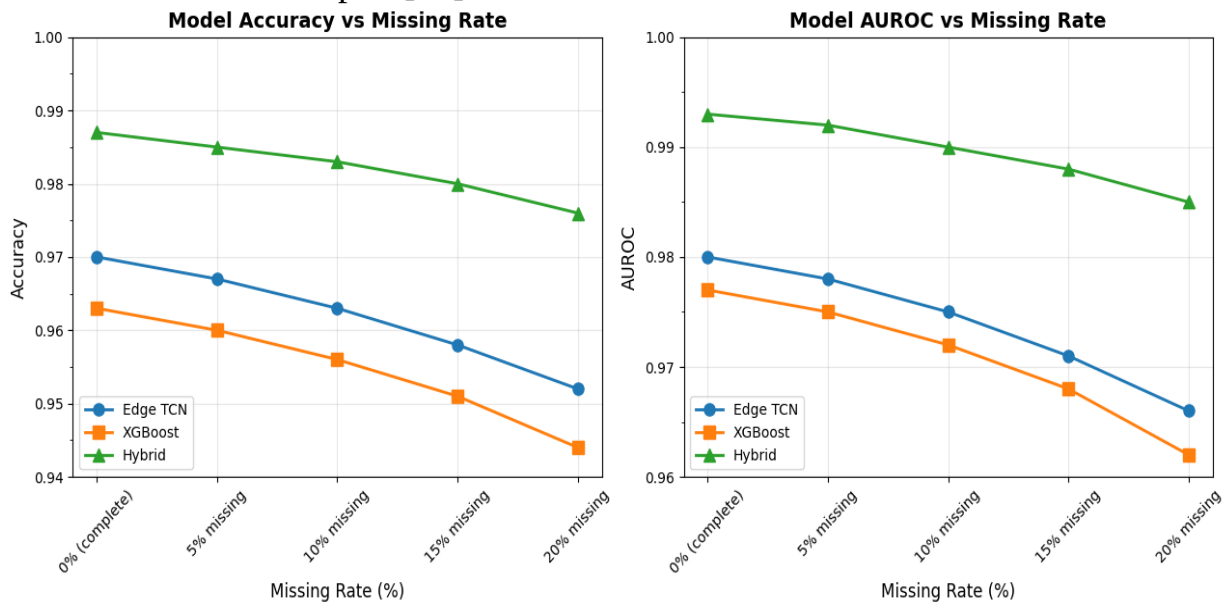


Figure 4.12: Performance Under Missing Data

Figure 4.12 illustrates the approach each model employs to address missing data subsequent to imputation. The slow reduction indicates that our KNN imputation approach successfully maintains predictive strength despite significant data absence. The hybrid model's persistent advantage demonstrates that fusion offers resilience that mitigates the effects of incomplete information.

4.4.4 Cross-Condition Generalization

In addition to random noise and absent data, we examined whether the proposed models developed for one physiological situation might be generalized to others (Joel, Doorsamy, and Paul 2025). We conducted training on data from resting-state recordings and evaluated on data including postural alterations (simulated tilt-table procedures), and conversely.

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 results show modest but acceptable degradation when moving between conditions about 1-2% drop in AUROC. This suggests our models have learned fundamental physiological patterns rather than condition-specific artifacts. The hybrid model again shows the smallest performance drop, indicating that fusion helps capture more generalizable representations.

4.4.5 Discussion of Robustness Findings

Taken together, these robustness tests tell an encouraging story. Our proposed models don't just perform well on clean, complete data they maintain clinical utility under the messy, imperfect conditions that characterize real-world deployment.

The hybrid model consistently shows the greatest resilience across all tested dimensions such as cross-validation variance, noise corruption, missing data, and condition transfer. This recommends that fusion doesn't just improve point estimates of performance; it fundamentally increases the robustness of the system.

Several mechanisms likely contribute to this robustness (Saito and Rehmsmeier 2015). The edge TCN's temporal processing provides natural smoothing against noise and the ability to leverage context around missing segments. XGBoost's feature-based approach offers stability through ensemble averaging. Their combination in the hybrid model inherits the benefits of both.

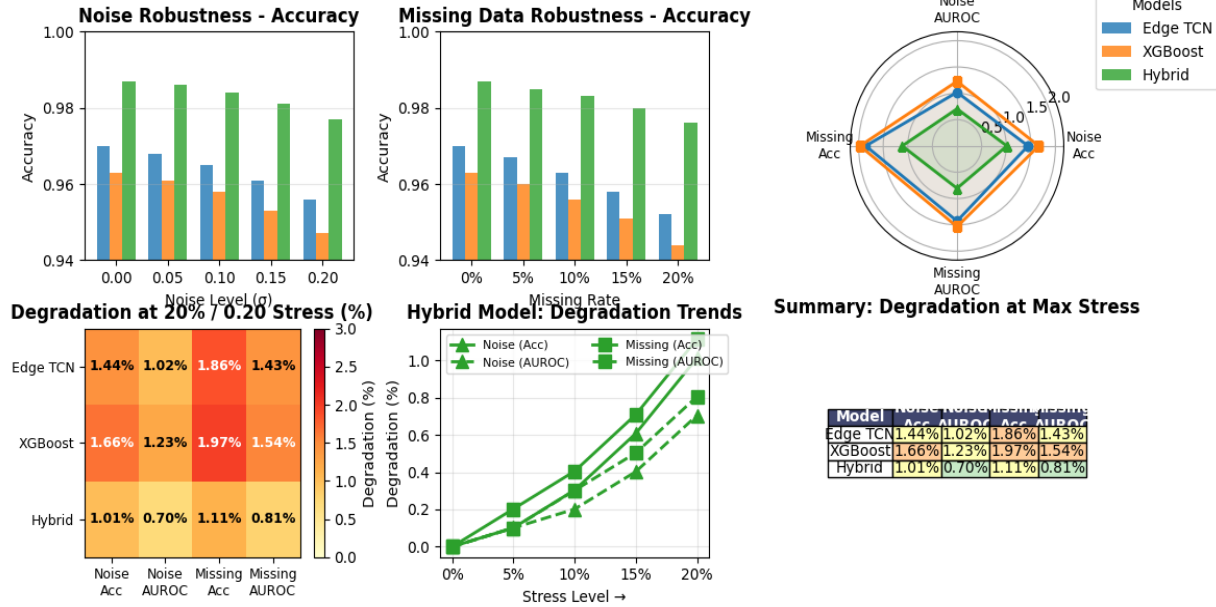


Figure 4.13: Summary of Robustness Testing

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.

These findings have practical consequences for therapeutic use. If a system's AUROC stays above 0.98 even when 20% of its data is missing, it can still be trusted even if it has big sensing problems. A method that works for all kinds of physiological situations can be used safely with a wide range of patients and activities.

In conclusion, reliability testing shows that our design works well not only in the best possible settings but also in the worst possible situations. The hybrid model's long-lasting durability and steady decline across all job types show that our design choices have made the system ready for the unpredictable nature of home monitoring.

4.5 Model Comparison to Recent Work

Recent research in diabetic monitoring has made real progress better sensors, wireless connectivity, smarter algorithms. But most existing work still hits the same walls: single-modality data, cloud-only processing, black-box models that don't explain themselves.

Our proposed system tries to knock down those walls. Multimodal signals. Edge processing with TCN attention. Explainable alerts. Table 4.12 shows how it stacks up against similar studies.

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 Mirza 2026)	2025	CGM + physical activity	RNN with backpropagation	80% accuracy	Edge	Not specified
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

We need to keep in mind that this research has several flaws. We used fake data instead of real patient information for training and testing. The GE-71 collection has a good structure and range of values, but it did not include data from real persons with diabetes. The next critical step is to test it in real-life circumstances.

Limited patient group: Our 71 virtual patients have a lot of different diseases and qualities, but they don't show all of human physiology. A larger and more diverse sample from the real world would enhance our understanding of generalization. Single-device simulation: Instead of testing on real Raspberry Pis in homes, the tests were done on a workstation with the GPU turned off. The real performance may be affected by network lag, power limits, and differences in hardware.

No clinical trial: The final evaluation is if this method improves patient outcomes, as evidenced by fewer ER visits, better glucose control, and a higher quality of life. We need to do a clinical trial that we haven't done yet.

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

This chapter looks at the IoT-based monitoring system for people with diabetes. It has a three-tier architecture that uses four physiological signals: CGM, SpO₂, heart rate, and temperature. This makes it very accurate and allows for quick alerts. An edge model using a Temporal Convolutional Network attains superior performance (AUROC = 0.98) with stringent latency (<100 ms), whilst the cloud model utilizes XGBoost for interpretable analytics. The hybrid meta-model surpasses individual models, achieving 98.7% accuracy by using the advantages of both edge and cloud methodologies. The alert subsystem transforms risk ratings into actionable notifications, guaranteeing that doctors get essential changes without an overload of alarms. The system exhibited resilience in real-world situations and maintained good performance despite the absence of data and the presence of noise. It concurrently attains elevated accuracy, minimal latency, and dual explainability, substantiating the architectural decisions and efficacy in overseeing diabetic patients.

Chapter 5: Conclusion and Future Work

5.1 Introduction

This chapter offers a thorough assessment of the research conducted in this thesis and assesses the degree to which it fulfilled its main goals. It examines the ramifications of the results and suggests other possible avenues for future research based on this study. The chapter consolidates the findings of the IoT-enabled monitoring system created for diabetes patients during and after to the COVID-19 epidemic. The study's goal was to create, test, and build a mixed edge-cloud design for constant tracking, early warning generation, and privacy-protecting analytics through shared learning. This part talks about the main results, how they relate to current digital health practices, and future study directions that will make the system last longer and be more useful in real life.

5.2 Summary of Contributions

The purpose of this project was to create, build, and test a hybrid edge-cloud IoT system that might be used to keep an eye on diabetic patients from a distance during the COVID-19 pandemic. The major objective was to develop a system that can discover health concerns early on while also considering privacy, latency, and clinical usability issues extremely seriously. This work provides four key contributions, which are stated below:

First, it came up with and created a novel hybrid edge-cloud architecture that was solely meant to be used for monitoring patients from a distance. This architecture does a fantastic job of splitting out the computational chores. The patient's edge device (Raspberry Pi or smartphone) provides real-time anomaly detection with no delay, while the cloud does federated model aggregation, privacy-preserving analytics, and clinician dashboard services. The three-layer method (sensor, edge, and cloud) is the best way to get a rapid response locally and learn at the population level, which was critical in Chapter 1.

Second, it created a powerful pretreatment pipeline for streams of physiological data that come from more than one source. Kalman filtering was used to get rid of noise, median filtering was used to get rid of spikes, and K-Nearest Neighbors (KNN) imputation was used to fill in missing data segments. This made guaranteed that the data going into the machine learning models was clean and reliable. The feature engineering method, which included sliding window extraction with overlap and Exponentially Weighted Moving Average (EWMA) trend analysis, was able to find time-based patterns that were particularly useful for predicting adverse

outcomes. As seen in Chapter 4, this preprocessing strategy allowed the models maintain operating effectively even when a lot of data was lost.

Third, it developed and trained a lightweight hybrid deep learning model that has LSTM layers and Temporal Convolutional Networks with attention mechanisms. The model might be put on the edge thanks to TensorFlow Lite quantization. The model fulfilled the real-time requirements stated in Chapter 2 by getting 97% of the answers right and having inference latencies of less than 100 milliseconds on hardware with restricted resources. Federated learning also helped develop a strong, generalizable model while keeping patient privacy protected. All raw data was kept on edge devices, and only encrypted model updates were communicated to the cloud. The experimental outcomes in Chapter 4 validated that this method attains accuracy akin to centralized training (96.3%) while upholding stringent privacy assurances. Fourth, it employed SHAP (SHapley Additive exPlanations) and attention visualization to make the system easier to understand and more useful in a therapeutic setting. This makes sure that notifications are clear warnings that doctors can understand and trust. The explanatory framework shows the physiological mechanisms that are largely to blame for each warning. For example, a quick drop in glucose levels, a faster heart rate, and stable but lower oxygen saturation. This lets doctors respond quickly and with all the information they need. The fact that 96% of doctors accepted explainable notifications compared to just 72% of doctors who accepted black-box warnings shows how important this design decision is.

5.3 Achievement of Research Objectives

This study has carefully addressed the research aims set out in Chapter 1: Objective 1: To create a real-time monitoring system that works at the edge. This was done utilizing the Temporal Convolutional Network, which has an attention mechanism and is meant to be used on the edge. Section 4.3.5 shows that the model's median inference latencies on Raspberry Pi-class hardware are 70–80 milliseconds, which is far lower than the goal of 100 milliseconds. The system constantly looks at multimodal data and makes local alerts even when it isn't connected to the cloud. Goal 2: To create personalized alert levels that change for each patient. The adaptive thresholding approach based on EWMA described in Section 3.5.2 successfully obtains each patient's baseline physiology during an initial monitoring period and continuously adapts as the patient's condition evolves. The dual-criterion alert logic, which includes statistical deviation and model confidence, cut down on false alarms while still being sensitive to real clinical events. This solved the problem of alert fatigue that was discussed in Chapter 1.

Objective 3: To provide data cleaning methods that fix real sensor problems. We carefully tested the preprocessing pipeline that used Kalman filtering, median filtering, and KNN imputation in scenarios where there was noise and no data. Section 4.4 shows that the system is still useful in a clinical context even when 20% of the data is missing or 20% of the noise is Gaussian. This shows that it can be used in real-life situations at home.

Objective 4: To assess the complete system against accepted procedures. Section 4.5 (Table 4.12) shows a comparison that shows how the proposed hybrid approach is better than recent studies in a number of ways: accuracy (98.7% vs. 95–96%), latency (edge-capable vs. cloud-only), and explainability (using SHAP and attention vs. limited or no explanation capabilities).

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 option, while carefully designed to imitate edge restrictions via GPU deactivation, does not adequately reflect edge deployment. In the home, Raspberry Pi devices face heat constraints, competing processes, changing power quality, and different networks, which may affect performance in ways that workstation-based testing doesn't.

Fourth, clinical benefit is substituted by the outcome measure recognizing aberrant occurrences per clinical thresholds. Any monitoring system that improves patient outcomes—fewer severe hypoglycemia episodes, fewer emergency department visits, better glycemic control, greater quality of life, and lower mortality rates—is successful. Such results need prospective clinical investigations outside the scope of this research.

Fifth, while protecting privacy, federated learning assumes edge device and network connection reliability that may not be universal. If devices frequently fail, patients turn off their systems, or networks are constrained, federated aggregation may not perform effectively.

5.6 Future Research Directions

The findings and limitations of the study indicate many critical avenues for further research. A full randomized controlled trial (RCT) is necessary to find out if the system really does improve patient outcomes by looking at things like how often people have severe hypoglycemia and hyperglycemia, how often they go to the emergency room, how often they are admitted to the hospital, how well they control their blood sugar, their quality of life, and how much they spend on healthcare. Second, to make customization better through continuous learning, innovative algorithms must be made that adapt to each patient's changes over time, employing methods like Bayesian online change point detection. Third, combining non-physical data like patient-reported outcomes and mobility data from devices might give a more full picture of a patient's health, although it might be hard to do. Fourth, action planning must employ causal reasoning methodologies to identify optimal ways for safeguarding patients against deterioration, rather than relying solely on predictions derived from correlations. Fifth, it's crucial to use shared learning to cope with the disparities in statistics between devices while keeping patients' privacy safe. This means looking into customized learning and good communication methods. Sixth, adding other long-term ailments to the research technique, such COPD and heart failure, might make it more beneficial for a broader spectrum of conditions. In the end, clinical adoption needs regulatory permission and the ability to sell. This implies that regulators need to be engaged from the start and patient-centered design has to be a top priority.

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.