Machine Learning–Based Non-Invasive Glucose Monitoring via Dual-Signal Acquisition

By

NAIM RASHID
Student ID: 21310009

ABU ANAS ANTAR
Student ID: 22121112

RIFAH TASPIA MITEE
Student ID: 22121027

MAYEESHA ISLAM
Student ID: 22121061

TASHRIK IBRAHIM
Student ID: 22121035

A Final Year Design Project (FYDP) submitted to the Department of Electrical & Electronic Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical & Electronic Engineering (BSEEE)

Department of Electrical & Electronic Engineering

BRAC University

May, 2026

Machine Learning–Based Non-Invasive Glucose Monitoring via Dual-Signal Acquisition

By

NAIM RASHID
Student ID: 21310009

ABU ANAS ANTAR
Student ID: 22121112

RIFAH TASPIA MITEE
Student ID: 22121027

MAYEESHA ISLAM
Student ID: 22121061

TASHRIK IBRAHIM
Student ID: 22121035

A Final Year Design Project (FYDP) submitted to the Department of Electrical & Electronic Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical & Electronic Engineering (BSEEE)

Academic Technical Committee (ATC) Panel Member:

Nahid Akhter Jahan, PhD (Chair)

Associate Professor, Department of EEE, BRAC University

Farjana Akter Jhuma, PhD (Member)

Assistant Professor, Department of EEE, BRAC University

Mr. Subhan Zawad Bihan (Member)

Lecturer, Department of EEE, BRAC University

Department of Electrical and Electronic Engineering
BRAC University
May, 2026

Declaration

It is hereby declared that

1. The Final Year Design Project (FYDP) submitted is my/our own original work while completing degree at Brac University.
2. The Final Year Design Project (FYDP) does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The Final Year Design Project (FYDP) does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I/We have acknowledged all main sources of help.

Student's Full Name & Signature:

NAIM RASHID
21310009

ABU ANAS ANTAR
22121112

RIFAH TASPIA MITEE
22121027

MAYEESHA ISLAM
22121061

TASHRIK IBRAHIM
22121035

Approval

The Final Year Design Project (FYDP) titled “Machine Learning–Based Non-Invasive Glucose Monitoring via Dual-Signal Acquisition” submitted by

1. Naim Rashid (21310009)
2. Abu Anas Antar (22121112)
3. Rifah Taspia Mitee (22121027)
4. Mayeesha Islam (22121061)
5. Tashrik Ibrahim (22121035)

of Spring, 2026 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in EEE (Bachelor of Science in Electrical and Electronic Engineering on 30th April of 2026. .

Examining Committee:

Academic Technical Committee (ATC): (Chair)

_____

Nahid Akhter Jahan, PhD
Associate Professor
Department of Electrical and Electronic Engineering
BRAC University

Final Year Design Project Coordination Committee: (Chair)

_____

Mirza Rasheduzzaman, PhD
Associate Professor
Department of Electrical and Electronic Engineering
BRAC University

Department Chair:

_____

Md. Mosaddequr Rahman, PhD
Professor
Department of Electrical and Electronic Engineering
BRAC University

Ethics Statement

We, the undersigned students (Naim Rashid, 21310009; Abu Anas Antar, 22121112; Rifah Taspia Mitee, 22121027; Mayeesha Islam, 22121061; Tashrik Ibrahim, 22121035) hereby confirm that our submitted work has been through proper plagiarism checks with reliable plagiarism checking software and the similarity percentage is significantly below 35%. We also declare that the material contained is the result of our joint work and any reliance on external sources and references has been adequately cited in an academic style. We consistently adhered to ethical guidelines in conducting the research and using the technology, such as obtaining informed consent from the volunteers, ensuring data privacy and security, transparency in the development of AI models, ensuring the well-being of the participants, and responsible use of machine learning methods. Furthermore, we continue to strive to reduce potential risks, maintain participant autonomy, and meet relevant professional and regulatory requirements, such as ISO 13485 for medical device quality management, and the Digital Security Act for secure data protection and handling.

Abstract / Executive Summary

Blood glucose monitoring remains a critical component of diabetes management, yet the discomfort and compliance challenges associated with conventional finger-prick methods have long prompted researchers to explore less invasive alternatives. This project comes in response to that sustained demand with a dual-sensor signal acquisition and machine learning-based estimation system that will be a potentially more accessible and patient-friendly way to monitor glycaemia, without requiring invasive blood sampling.

The system is based on two different and complementary sensing modalities: Near-Infrared (NIR) spectroscopy and Galvanic Skin Response (GSR). The light transmitted through NIR reflects the biochemical composition of glucose molecules in the subcutaneous tissue, whereas the changes in GSR are related to the electrodermal changes related to the physiological fluctuations associated with glucose dynamics. The dual-acquisition system is not meant to use either signal alone but in an attempt to enhance the robustness of the estimation by utilizing the complementary information contained in each.

A custom dataset collected through extensive fieldwork and on-site data acquisition is at the core of the system's functioning, designed to reflect the natural variability in physiological data. It is generally believed that training a machine learning model using domain-specific data of this type is more reliable than adapting generalized benchmarks, especially for glucose estimation due to the sensitivity of these estimations to individual biological variations. The model learns how to relate the output of the various sensors to the reference glucose values and is able to predict glucose values that are based on empirical patterns rather than purely theoretical proxies.

The system is implemented to be constantly monitored, and it can be viewed via a web-based interface, without the need for repeated invasive procedures to monitor glucose trends. Although additional clinical testing will be needed before it could be introduced clinically, these results indicate that this sensor fusion method has significant potential as a non-invasive alternative, especially for people who need to test glucose levels more often.

Keywords: *Non-Invasive Glucose Monitoring, Machine Learning, Biomedical Signal Processing, Galvanic Skin Response (GSR), Near-Infrared Sensor.*

Acknowledgement

We would like to sincerely thank our respected ATC panel members Dr. Nahid Akhter Jahan, Dr. Farjana Akter Jhuma, and Mr. Subhan Zawad Bihan for their valuable guidance and continuous support throughout this project. We are also deeply grateful to our parents for their encouragement and understanding, which kept us motivated during this journey.

Table of Contents

Chapter 1: Introduction- [CO1, CO2, CO10]

1.1 Introduction

Diabetes mellitus is among the most pressing chronic disease challenges of the twenty-first century. Defined by persistently elevated blood glucose concentrations, the condition is deceptively silent in its early course yet profoundly damaging over time, progressively compromising renal, cardiovascular, and retinal function in affected individuals [1]. The scale of the problem is difficult to overstate: data compiled by the International Diabetes Federation indicate that roughly one in nine adults currently lives with the condition, and projections suggest this figure will reach 853 million within two decades [2]. Against this backdrop, the routine measurement of blood glucose is not merely a clinical convenience but a medical necessity, one that determines whether the disease is managed effectively or allowed to erode quality of life undetected.

Conventional glucose measurement relies almost exclusively on methods that require either a capillary blood sample, obtained via finger-prick lancets, or the subcutaneous insertion of a sensor filament as in continuous glucose monitoring (CGM) systems. Both modalities carry an inherent burden: they are painful, carry a non-trivial infection risk, and impose a compliance cost that is especially prohibitive for patients who require multiple tests per day [3]. For children with autism spectrum disorder who also present with diabetes, and for individuals with needle phobia more broadly, even a routine blood draw can precipitate acute anxiety, sensory overload, and, over the longer term, a sustained behavioral resistance to monitoring that clinicians struggle to address [4]. The aggregate consequence is a monitoring deficit: patients test less frequently than clinically indicated, glycaemic excursions go undetected, and the cumulative burden of uncontrolled diabetes compounds.

It is within this clinical and technological gap that the present project is situated. Non-invasive glucose estimation using optical and electrodermal sensing modalities has attracted sustained research interest over the past two decades, yet no device in this category has achieved regulatory approval from the U.S. Food and Drug Administration. The reasons are instructive: single-modality approaches, whether based on near-infrared (NIR) spectroscopy, photoplethysmography (PPG), or galvanic skin response (GSR) alone, have consistently failed to deliver the combination of accuracy, reproducibility, and robustness under real-world conditions that clinical validation demands [5]. Sensor fusion, the integration of signals from multiple complementary modalities, has emerged as the most promising avenue for closing that gap, though prior multi-sensor implementations have often involved complexity and cost that undermine their practical viability [6][7].

This project proposes a pragmatic middle ground: a dual-sensor non-invasive glucose monitoring system that combines near-infrared spectroscopic measurement with galvanic skin response sensing, coupled with server-side machine learning inference and a web-based interface for longitudinal trend analysis. The dual-sensor architecture is premised on the principle of data diversity, wherein two sensors with complementary noise profiles and low

inter-modal correlation produce estimation errors that partially cancel upon fusion, yielding superior accuracy relative to either sensor deployed in isolation.

1.1.1 Problem Statement

While it is well established that patients need to monitor their blood glucose level regularly to manage diabetes, existing tools for this monitoring are still intrusive, painful and in some cases unattainable for patients. There is no FDA-approved alternative to invasive testing, and the difference between what has been technically shown to be possible in research studies and what is necessary for FDA approval is significant. To date, single-sensor non-invasive methods have not achieved the accuracy necessary for clinical use, and multi-sensor systems that have achieved good clinical-level accuracy are often too expensive and complicated to be deployed in low resource settings [5][7].

The sub-optimal care of the most vulnerable population groups is not being addressed in the meantime. The current monitoring gap is not met by existing technologies, whether a child has co-occurring diabetes and autism, has needle phobia, or is in a low-to-middle-income country, like Bangladesh, where invasive testing is the only available approach. A local survey of undergraduate students at BRAC University showed that 97.1% of all those students who had had a test done for glucose had done it by some method that necessitates a blood draw, indicating that non-invasive testing is essentially non-existent in the local health system. The issue is not just technical: a lack of an affordable, non-invasive, clinically reliable glucose monitoring device is a systemic barrier to equitable diabetes management.

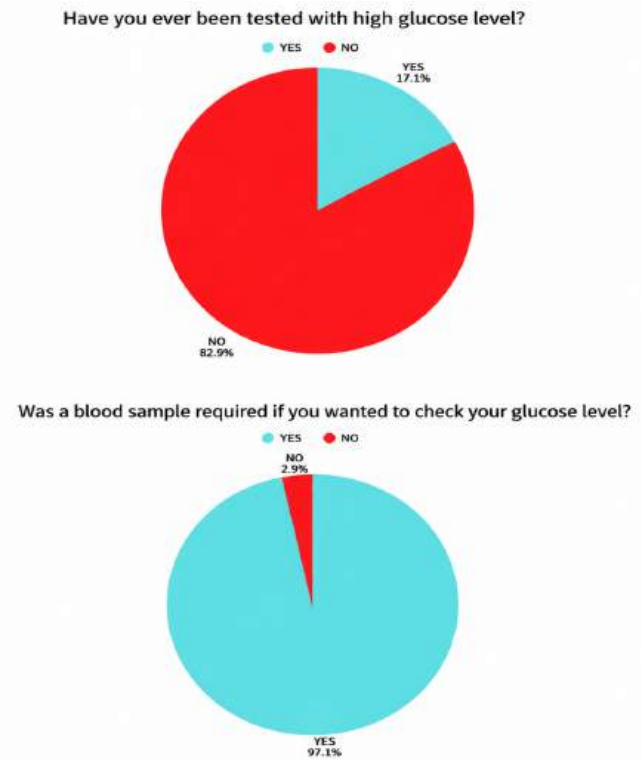


Figure 1.1: Survey on Blood Glucose Level Monitoring Among BRAC University Undergraduate Students.

1.1.2 Background Study

Non-invasive optical glucose sensing is based on the ability of blood and interstitial tissue to absorb near-infrared light differently, depending on the glucose concentration. Glucose has characteristic absorption characteristics within the NIR spectrum that in principle could be used to infer its concentration from the attenuation of an optical signal transmitted or reflected in the spectrum. A similar principle is used in photoplethysmography, which measures the change in optical absorption of arterial blood over the course of the cardiac cycle. In contrast, the galvanic skin response is an electrodermal measure of activity in the sympathetic nervous system, which controls the secretion of eccrine sweat glands. Physiological arousal states and metabolic perturbations induced by glycaemic variability modulate the output of sudomotor activity with predictive signals, it seems, but it is not a biochemical effect.

All of these have been studied separately in the non-invasive glucose literature with mixed findings. The technique with the best direct correlation to glucose concentration measured in near-infrared spectroscopy, but is influenced by the wide inter-individual difference in skin thickness, melanin content, tissue hydration, and optical scattering properties [5]. While the PPG-glucose relationship proved to be indirect and noisy, with uniformly negative R2 scores across various machine learning architectures applied to the Mendeley PPG dataset employed here [8], the relationship between PPG and other parameters demonstrated in this work has more direct and stronger predictive power. GSR has been applied only in psychophysiological studies and glucose estimation using the modality is still in exploratory phase, with multi-attribute datasets like GlucoBench indicating that the modality also contains complementary predictive information when used in conjunction with biochemical sensors.

The theoretical and empirical foundations for the concept of sensor fusion. This was verified by the Monte Carlo simulation carried out in the framework of this project, where it was found that the mean squared error when two sensors are used together is reduced when the two sensors have low mutual correlation in the error distribution (MSE = 21.93) compared with when the two sensors are highly correlated (MSE = 23.03). From an empirical point of view, multi-sensor systems from the literature have been shown to have more stable performance profiles than single-sensor ones, in spite of the fact that no optimal set of features could be found across these studies [7]. Together, these factors drive the pairing of NIR and GSR over the incorporation of more sensors or higher-order configuration because they offer cost, complexity of integration and diminishing returns for predictive return with more modalities of similar biochemical nature.

1.1.3 Literature Gap

The literature review shows that there are a number of overlapping gaps that serve as the motivation for the current research trajectory. First, although both NIR and GSR-based glucose monitoring have been considered as separate sensing systems, the joint use of these two types of sensors in a time-synchronised dual acquisition system is not well documented. The studies of the

multi-sensor type have either used more than two sensors, or used two or more modalities that have similar noise characteristics, which restricts the diversity advantage sought by the present design. Second, there are no publicly available datasets that correlate synchronised NIR and GSR signals with clinically established glucose reference measurements: existing benchmarks, like the Mendeley PPG dataset and the Kaggle NIR dataset, focus on the individual moduli, and the GlucoBench dataset included a wealth of contextual physiological data but was not built around the particular sensor combination used here. Third, previous comparative research using machine learning has tended to concentrate on the overall regression results, without systematically investigating the computational feasibility of the model deployment on embedded and server architectures. That's significant – a model that performs well in an environment that leverages a massive data center could be very different from a model that performs well in an environment where it's being deployed on a microcontroller, and that difference isn't always explicitly addressed in the evaluation methodology. In the present work, all three gaps are addressed by developing a tailored (synchronized) dataset, comparing models using both a lightweight and a deep learning architecture, and using an explicit decision rule to choose which deployment paradigm to use based on empirical performance evidence and hardware suitability criteria.

1.1.4 Relevance to current and future Industry

The validated non-invasive glucose monitor has a significant commercial and clinical application. The global blood glucose monitoring market is one of the largest segments of medical device manufacturing and the majority of the products are essentially the same as the invasive products which were introduced years ago, with the finger-stick glucometer as the primary product. The lack of an FDA approved non-invasive device is not so much an indication of non-commerciality, but it is an indication of a stubborn accuracy challenge that the industry has yet to overcome. Any system able to get close to the ±10 mg/dL accuracy limit that was set in this project and which was tested against the clinical standard from the Clarke Error Grid would be a step in the right direction.

Structural constraints in the healthcare sector further amplify the relevance in the Bangladeshi context. The country has a significant and increasing diabetes burden, and access to CGM is restricted due to cost and chain of supply issues. An expected bill of materials ranging from BDT 16,500 to 29,000, and based on a microcontroller-based platform and not on proprietary sensor cartridges, might seem viable to local medical technology players to manufacture and service. By moving the heavy compute footprint to scalable cloud infrastructure, the server-based inference architecture also moves the cost of the hardware to the cloud, eliminating costs at the device level, which is increasingly possible even in bandwidth-limited scenarios. Industry-wide, the project falls under three emerging trends: miniaturization of wearable biosensors, increased server-side deep learning inference pipelines, and a shift towards policy focus on chronic disease prevention and management by patients.

1.2 Objectives, Requirements, Specification and Constraint

1.2.1. Objectives

The project will tentatively aim to incorporate the following objectives:

- Implement a dual-sensor system to estimate an individual's blood glucose level.

- Create a custom, time-aligned dual-sensor dataset and use it to train models that map raw inputs to reference glucose values.
- Design and develop a web-based interface to deliver long-term glucose trend analytics and provide clinical decision support.

1.2.2 Functional and Nonfunctional Requirements

Functional Requirements:

Table 1.1: Functional Requirements.

Feature	Description
Sensor Integration	Capture real-time bio-signals using dual sensors
Data Acquisition	Collect synchronized sensor data with a timestamp
Signal Processing	Extract features from sensors for glucose estimation
ML Prediction Model	Map raw sensor data to glucose levels using trained ML algorithms
Interface	Display glucose readings on the web dashboard
Data Storage	Store the user's historical glucose data for trend analysis

Non-functional Requirements:

Table 1.2: Non-Functional Requirements.

Feature	Description
Portability	Must be small and portable
Power	Run sufficiently long enough to operate in clinical conditions using battery
Latency	Display results within a few seconds after measurement
Accuracy	Comparable results with reference glucose levels (Finger Prick Test)
Safety	Must be non-invasive and comply with applicable standards and codes.
Usability	Should be easy for patients to use independently, without requiring assistance from a healthcare professional.
Cost	Should be affordable for low to middle-income users.

1.2.2 Specifications

- It can be used for a long time in the clinic or at home, because the device can last for 6-8 hours after charging once.
- The system provides results in 5-10 seconds since the finger is pressed down, with real-time signal processing and machine learning algorithms.
- The device measures glucose with a goal of ±10 mg/dL when using finger prick test results.
- Compact and handheld (approx. 8 x 6 inches) device suitable for non-clinical, home use.
- The instantaneous glucose reading is shown on a 16x2 alphanumeric LCD display, and is also uploaded to a web interface for long term analysis.
- To ensure safety, emitted light power is limited to less than 50 mW/cm2, which does not cause damage or discomfort to the skin.

1.2.3 Technical And Non-Technical Consideration And Constraint In Design Process

The overall design process involved a combination of technical and non-technical constrains. These constraints are discussed separately below.

Technical Constraints: These are the set of constraints which are imposed by the external factors in the technical domain which the design has to fit into.

- The inter-individual variability in near-infrared transmittance through the fingertip, which varies more significantly with skin tone, melanin density and tissue thickness, can affect the model estimate and cannot be completely overcome by signal conditioning alone, potentially resulting in some bias in model estimates for individuals not well represented in training data sets.
- The low frequency noise components from finger motion during acquisition are mixed with the signal bandwidth of the physiologically relevant signal and need to be properly rejected in real time at the signal conditioning stage, with robust artifact rejection.
- The NIR photodiode can be saturated in some lights, so optical shielding is needed in the sensor housing to get a good signal.
- The GSR signal can be contaminated by external electromagnetic interference, especially the 50 Hz power line noise and the muscle activity artifact, which necessitates a bandpass filter limiting the analysis to the range of 0.5 to 5Hz for physiologically relevant GSR changes.
- The GSR signal after amplification is bipolar, while the microcontroller ADC only supports unipolar signals between 0 and 5 V, thus necessitating a separate level-shifting offset stage in the analog front-end.
- The machine learning models trained with a small or demographically limited data set may lack generalisation properties to unseen people, which is especially relevant when the diversity in the physiology of the humans is well known and is a confounder for a sensing modality.

- The latency dependency on network connectivity that the server-based inference architecture brings makes it hard to guarantee the five-to-ten-second response time in environments with poor network stability and connection quality, especially when using wireless networks.
- Complex deep learning models are not appropriate for on-device inference due to limited memory and processing power offered by embedded hardware platforms like ATmega32, and call for a server-side architecture to achieve high accuracy estimation.

Non-Technical Constraint: These are the external (non-engineering) constraints and barriers that affect the context and nature of the project.

- Because of the sensitive nature of the biometric and health data being gathered, dataset construction must be approved by the University of Guelph ethics board and the informed, written consent of all participants is required.
- Data of all participants should be anonymised and kept in line with the relevant data protection laws and regulations, such as GDPR and equivalent privacy laws in Bangladesh.
- A logistical constraint of the academic project is to avoid oversize and demographically unrepresentative samples which limit the extent to which models can be trained and validated within the project time.
- If the device is to be applied to any clinical use outside the academic setting, it will need premarket validation according to FDA or comparable regulatory requirements, which is beyond the scope of the present project but which would be considered in documenting and in testing throughout.
- The project budget limits the number of components that may be chosen, especially for the sensing subsystem, where more accurate FDA registered sensor alternatives might be found but at a cost that is not compatible with the BDT 16500 to 29000 target cost.
- The system is intended to provide information for health care decisions, and as such has an ethical responsibility to not overstate the accuracy or clinical reliability of any information provided in reports, presentations or future publications, as required by the transparency requirements of the NSPE Code of Ethics.
-

1.2.4 Applicable compliance, standards, and codes

The design is guided by several applicable standards and regulatory frameworks, represented in Table 1.3.

Table 1.3: Applicable Standards and Codes

Standard	Title	Applicability
ISO 14971	Application of risk management to medical devices.	Essential for identifying hazards, risks, and control measures.
ISO/IEC 27001	Information security management.	Critical for cloud-based systems handling personal health data.

IEEE 11073	Personal Health Device Communication Standards	For health-related data exchange and integration with apps/cloud
IEEE 7000 series	AI Ethics and Standards (e.g., 7001, 7002, 7003)	For ethical use of AI in medical diagnostics, privacy, algorithm bias, and transparency
IEC 60601-1	Medical electrical equipment – General requirements for basic safety and essential performance	Core safety standard for all electrical medical devices
IEC 60601-1-11	Requirements for medical devices in the home healthcare environment	If device is intended for personal/home use

1.3 Systematic Overview/summary of the proposed project

The proposed system is an end-to-end solution including signal acquisition with hardware, analog signal conditioning, wireless cloud communication, machine learning inference on the server and visualization on the web. The hardware level features two sensors attached to an enclosed fingertip. The sensors used are off the shelf sensors, the MAX30105 and GSR. The signal acquisition is performed by an Arduino-based platform and the cloud communication is done using an ESP32 module, which is only used for cloud communication.

The NIR front-end incorporates a transimpedance amplifier (TIA) to convert the current from the photodiode to a voltage appropriate to the ADC, followed by a second op amp stage of gain, based on the 741 integrated amplifier, for additional gain. The GSR front-end features a Wheatstone bridge detection stage biased at 0.5 V to ensure linearity of the skin conductance signal, an AD620-based differential preamp with programmable gain, an active bandpass filter (0.5 to 5 Hz) to suppress noise, and a TL081-based non-inverting amplifier with a level-shifting offset stage to convert the bipolar output to the unipolar 0–5 V range accepted by the microcontroller's ADC. The sensor channel signals are digitised and analysed by the arduino platform, and predefined features are extracted in the time domain and sent to the cloud server by the esp32 for remote inference.

We created a machine learning pipeline to test a variety of regression models including CatBoost, XGBoost, LightGBM, Gradient Boosting, Random Forest, Extra Trees, Ridge Regression, Bayesian Ridge, ElasticNet, Support Vector Regression and K-Nearest Neighbors. They were assessed fairly by employing five-fold cross validation and assessing by MAE, RMSE and R^2. We also created ensemble models using stacking and voting techniques to make the predictions more stable and robust. On top of that, we used Optuna for hyperparameter tuning of the model automatically, particularly of CatBoost to squeeze out as much performance as possible, to minimize prediction errors, to help the model generalize to new data.

The project is divided into three main phases:There are three main phases of the project: The initial phase (FYDP-P) was devoted to developing a thorough understanding of the problem, extensive literature review and conceptualization and design. At present, the FYDP-D phase is ongoing, during which we have been conducting extensive testing and verification based on

simulations. This includes MATLAB Simulink to perform multimodal biosignal processing, Proteus to perform analog circuit validation, Cisco Packet Tracer to measure the feasibility of cloud communications, Google Colab to benchmark and optimize machine learning models, and Monte Carlo simulation to statistically compare redundancy-based and diversity-based sensor fusion approaches. For the final phase (FYDP-C), we will develop and test a physical breadboard prototype, gather our own real-world data from participants whose informed consent will be obtained, re-train our models on our project-specific data and conduct full end-to-end validation against the readings of standard reference glucometers.

1.4 Conclusion

This chapter has laid the bases for the clinical and technical building blocks on which the rest of the report is based. These factors: the global burden of diabetes, the limitations of current non-invasive diabetes monitoring, and the groups of people most at risk of the current diabetes monitoring paradigms make the proposed research direction compelling. Previous studies have demonstrated that each of these sensing modalities has been well investigated, but only a few systems have incorporated the combination of NIR and GSR together in an architecture that would benefit from being cost-feasible and/or inference based on the server. The vision, specifications and constraints create a space that is both challenging and achievable, while the standards and ethical issues provide boundaries for good engineering practice.

The remaining chapters describe the various design approaches that were explored, the engineering tools used for design simulation and verification, and the validation results of the designs, which form the basis for a selection of Design Approach II as the best design architecture. In all of this, the analysis is based on empirical performance data and supported by the theory presented here, allowing for the technical justification of each design decision and in the context of the larger body of literature on non-invasive glucose monitoring.

Chapter 2: Project Design Approach [CO5, CO6]

2.1 Introduction

The clinical need and the technical limitations of the project have been set, the question that now needs to be answered is “how should the proposed non-invasive glucose monitoring system be designed”, and this is what the next step of the project involves. There is no straightforward answer to that question. There are several plausible designs that can be used; each design comes with a specific set of trade-offs between accuracy, cost, power consumption, latency, and maintainability. Intuition and convenience are not sufficient as a basis for selection from an array of choices when the device is to be used to make health care decisions. The chapter therefore follows a structured comparative approach, where several different design solutions are identified, described in a technical manner, detailed enough to make meaningful comparisons, and then analysed with regard to a set of common performance criteria, aiming at the selection of the best design for implementation and validation in the next step. The overall functional diagram is shown in Figure 2.1.

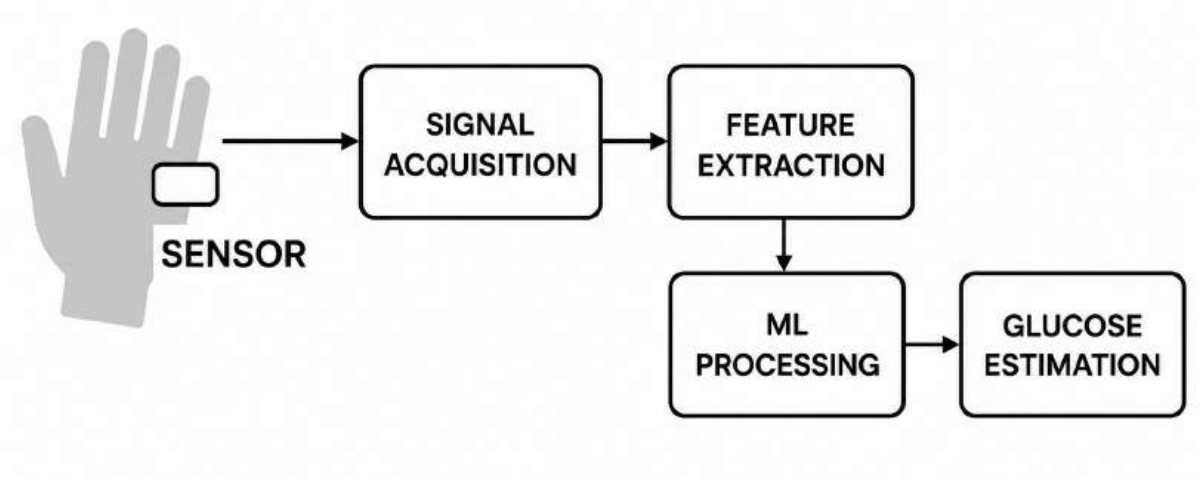


Figure 2.1: Overall functional diagram of a non-invasive glucose monitoring system.

There were two major design ideas. The high-level function is the same: They both require learning two physiological signals, then extracting discriminative features and then mapping these features to a continuous blood glucose estimate. The distinction is in the sensor strategy and the computational architecture that is used, with implications for nearly all other aspects of the system. The first one uses the same type of sensor in both channels and applies simple machine learning algorithms on the embedded sensor. The second approach is to use two separate sensors that measure two separate aspects of the glycaemic variation, and to shift the burden of inference to the cloud, where a more powerful machine can run the more computationally-intensive deep learning models than an embedded sensor.

We examine each of these three sets of evidence, namely machine learning benchmarking on a number of proxy datasets, a Monte Carlo simulation comparison of the statistical properties of the redundant dual versus diverse dual sensor fusion, and a comparative cost and power analysis of the two hardware configurations. All of these streams together form the basis of selection of a design that is not based on assumption, but on empirical data.

2.2 Identify Multiple Design Approach

The two design approaches evaluated in this chapter emerged from a systematic review of the architecture options available within the project's technical and financial constraints. Several candidate configurations were considered during the conceptual phase before being consolidated into two representative approaches that span the relevant design space.

At one end of that space, shown in Figure 2.2, lies the simplest possible dual-sensor architecture: two identical sensors operating in parallel on the same physical principle, with all computation performed on the device itself. This configuration, designated Design Approach I, maximises measurement redundancy and minimises communication complexity, but sacrifices sensor-level diversity and places the full inference burden on hardware that was not designed for intensive computation.

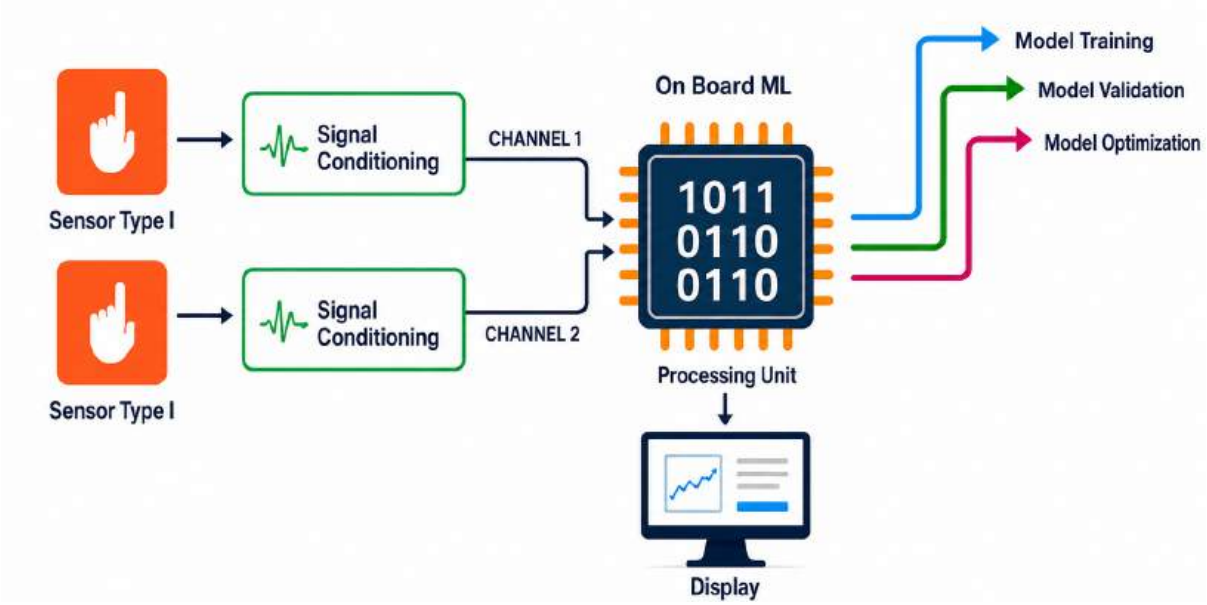


Figure 2.2: Block diagram of Design Approach 1.

At the other end, shown in Figure 2.3, lies a heterogeneous sensor pairing in which two physically distinct modalities contribute complementary information streams, with inference offloaded to a server or cloud backend capable of running the most accurate deep learning architectures available. This configuration, designated Design Approach II, sacrifices the simplicity of a self-contained device in exchange for greater predictive accuracy, lower device-side power consumption, and a maintenance model that allows the inference model to be updated centrally without physical intervention.

A third class of architectures, involving more than two sensors, was considered and set aside. While the literature suggests that multi-sensor fusion beyond two modalities can yield incremental accuracy gains, the associated increases in hardware cost, integration complexity, and data synchronisation burden were judged disproportionate to the expected benefit at the current stage of development. The two approaches described below therefore represent the most informative points of comparison within the feasible design space.

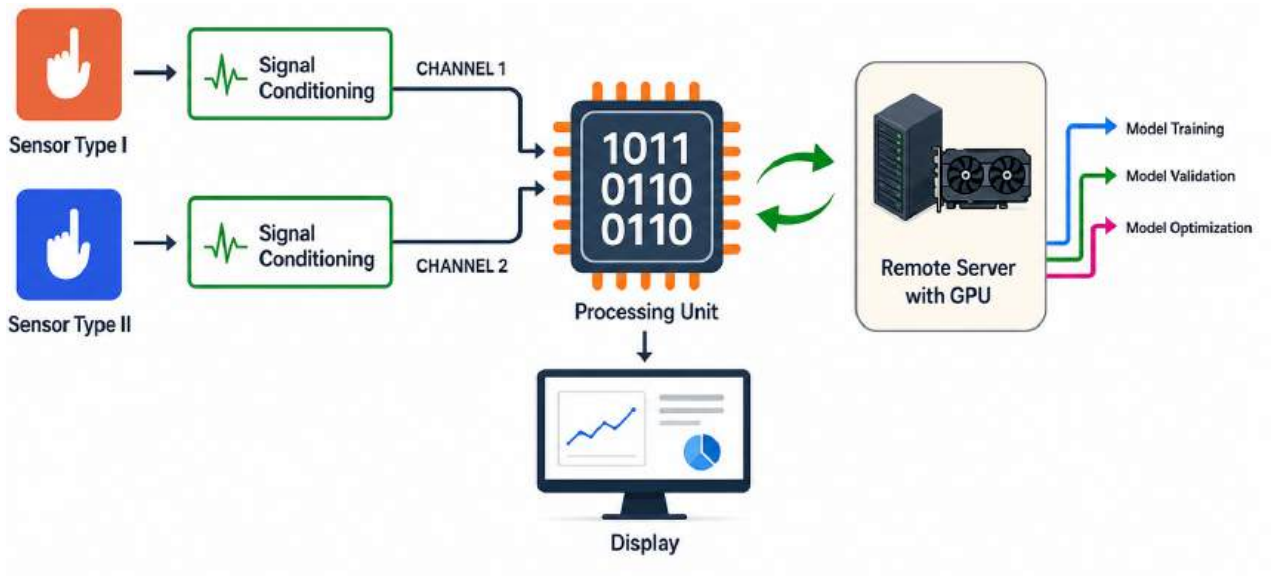


Figure 2.3: Block diagram of Design Approach 2.

2.3 Describe Multiple Design Approach

Design Approach I is achieved by using the same type of sensor in two parallel acquisition channels. The idea is one of redundancy - the physiological signal is sampled twice at the same time to average over channels to remove the effect of random measurement noise, transient artifacts, and to improve the stability of the feature vectors supplied to the inference model. The digitised outputs from both channels are combined at the processing unit and independent signal conditioning stages are provided between the channels.

In this setup, the processing unit is a high-performance single board micro-processor that can execute lightweight machine learning models on the device. The models used in this architecture are linear regression and shallow neural networks, which have only a few parameters and minimal memory requirements, and can be compiled and run on the computational capacity of embedded hardware. Model training and validation are done offline on a development machine and model weight is then flashed to a device where it is rigid until a physical weight update is done.

The key characteristics of this approach are as follows:

- Both acquisition channels use the same sensor type, prioritising redundancy and signal reliability over modality diversity.
- Machine learning inference is performed entirely on the device using lightweight models, eliminating the need for network connectivity during operation.
- Response latency is low, as there is no data transmission delay between the device and an external server.
- On-device computation draws continuously on the battery, resulting in higher power consumption relative to an architecture that offloads inference.
- Updating or retraining the model requires physical access to the device, making iterative improvement after deployment logistically demanding.

Design Approach II is based on the diversity of data rather than redundancy logic of Design Approach I. It does not sample the same physiological phenomenon twice, but it compares two sensors that measure two different physiological responses to glycaemic variation. The first sensor is a near-infrared spectroscopic transducer that will “read” signatures of optical absorption associated with glucose in the blood and interstitial tissue. The second is a galvanic skin response sensor that detects electrodermal activity resulting from sympathetic nervous system responses to metabolic and/or physiologic changes.

Each sensor channel is connected to a separate analog front-end: the NIR channel is combined with a transimpedance amplifier, a low-pass filter, and a secondary amplifier to amplify the current of the photodiode to within the ADC range of the microcontroller; the GSR channel is connected to a Wheatstone bridge biased to 0.5 V, an AD620 instrumentation amplifier, an active bandpass filter from 0.5 to 5 Hz, a non-inverting amplifier with a gain of ~100, and a level shifting offset amplifier to convert the bipolar amplified signal to a unipolar 0-5 V input for the ADC. Both feature vectors from each channel are sent through Bluetooth low energy to an attached device that then sends them over the internet to a Flask-based web server. The inference model is selected from the set of benchmarked deep learning architectures (such as Hybrid CNN-Transformer and Transformer) and provides a continuous glucose estimate, which is sent back to the device and shown on the screen.

The main features of this method are:

- The two different sensors are physically separated to measure complementary biochemical and electrodermal aspects of the glycaemic variation, and not redundant data.
- The deployment of Advanced machine learning models like deep neural networks and Hybrid architectures is executed on a remote server, access to GPU acceleration and large memory that allows to execute a level of predictive complexity that can't be provided by embedded hardware.
- Because of the data transmission round-trip between device and server, the response latency is higher than that of Design Approach I, but is acceptable under normal network conditions.
- This reduces power consumption on the device side significantly because the model inference is not done on the device, but instead is handled by the microcontroller, which is then responsible for data acquisition, feature extraction, and wireless transmission.
- The inference model can be centrally retrained and updated on the server, as new data is collected, with no re-flashing or changes to the firmware of deployed devices.

2.4 Analysis Of Multiple Design Approach

2.4.1 Machine Learning Performance Analysis

To evaluate the predictive accuracy that can be obtained with each design method, machine learning models typical of both types were trained and tested on three publicly available proxy datasets for the NIR, PPG, and GSR modalities. The metrics used for the evaluation were the coefficient of determination (R^2), mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE), where a lower MAE, MAPE, and RMSE and a higher R^2 value corresponded with better predictive accuracy.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\, y_i - \hat{y}_i \,| \quad \text{.....(Eq 2.1)}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \quad \text{.....(Eq 2.2)}$$

$$R^2 = 1 - \frac{\Sigma_i^N (y_i - \hat{y}_i)^2}{\Sigma_i^N (y_i - \bar{y})^2} \quad \text{.....(Eq 2.3)}$$

On the NIR dataset, shown in Figure 2.4, both the lightweight linear regression model and the lightweight CNN, representative of Design Approach I, performed well, achieving R^2 values of 0.9605 and 0.9662 respectively. This result reflects the relatively direct relationship between NIR voltage and glucose concentration in a low-dimensional dataset, where a simple model is sufficient to capture the underlying mapping. It is worth noting, however, that this dataset contains only one input feature and one output variable, which means the apparent strength of these results cannot be extrapolated to the more complex, multi-feature inference problem the deployed system will face.

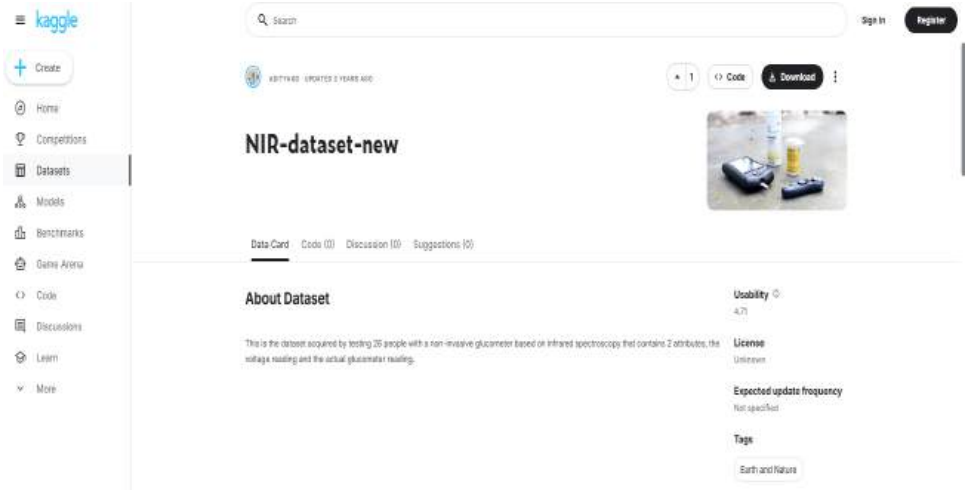


Figure 2.4: NIR Dataset (collected from Kaggle).

Comparison

Table 2.1: NIR Dataset ML Model Comparison.

ML Model	R^2 score	MAE	MAPE	RMSE
Linear Regression	0.9605	1.2341	1.43%	1.6001
Lightweight CNN	0.9662	1.3115	1.49%	1.4809

Ranking (in terms of R^2 score)

1. Lightweight CNN
2. Linear Regression

The PPG dataset, shown in Figure 2.5, told a markedly different story. All models evaluated, from linear regression through to ResNet34, returned negative R^2 scores, indicating that none could outperform a simple mean prediction. The best-performing architecture was the Hybrid CNN-Transformer, which achieved an R^2 of -0.0029 and the lowest MAE of 11.66 mg/dL, followed by the Transformer at -0.9013. These results are consistent with the established understanding that PPG signals carry only an indirect and noise-sensitive relationship with blood glucose concentration, and they provide a clear empirical basis for excluding PPG as one of the two selected sensing modalities.

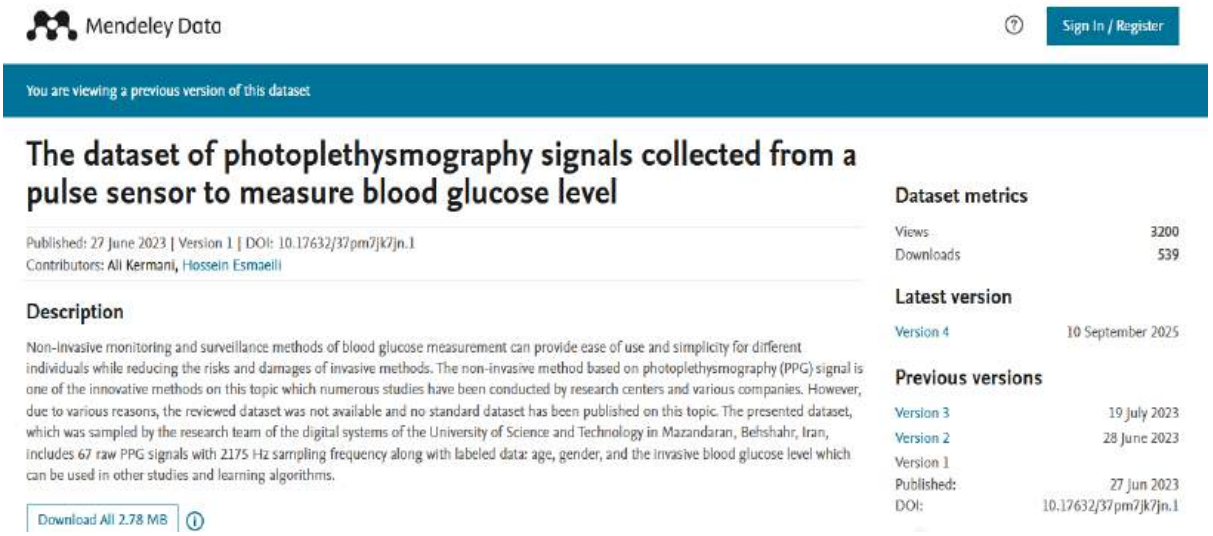


Figure 2.5: PPG Dataset (Collected from Mendeley)

The dataset consists of physiological and signal-derived parameters collected from subjects for the estimation of blood glucose levels. The features and their descriptions are as follows:

Table 2.2: Features Extracted from PPG Dataset.

Feature	Description
height	The height of the participant in centimeters. It provides a measure of body size, which can indirectly influence hemodynamics and PPG signal characteristics.
weight	The body weight of the participant in kilograms. Body composition affects blood flow, vascular compliance, and optical path length in PPG measurement.
ppg_mean	The mean value of the PPG signal amplitude over one or more cardiac cycles. Represents the average blood volume pulse intensity.

ppg_std	The standard deviation of the PPG signal amplitude, capturing signal variability due to physiological or motion changes
ppg_min	The minimum amplitude value of the PPG waveform in the analyzed segment. Reflects the trough point corresponding to diastolic blood flow.
ppg_max	The maximum amplitude value of the PPG waveform in the analyzed segment. Represents the systolic peak where arterial blood volume is highest.
ACDC_ratio	The ratio between the alternating current (AC) and direct current (DC) components of the PPG signal. A key feature for analyzing relative pulsatile strength and perfusion index; higher ratios indicate stronger pulsatile components.
rise_time	The time duration from the onset of the PPG pulse to its systolic peak. It represents how quickly blood is pumped during each heartbeat and is influenced by arterial stiffness and cardiac output.
decay_time	The time from the systolic peak to the return to baseline (or diastolic level). It relates to vascular resistance and venous return characteristics.
pulse_width	The width of a single PPG pulse at a given percentage of its peak amplitude (typically half). It represents the duration of the cardiac cycle and is inversely related to heart rate.
area_under_pulse	The integrated area beneath one PPG waveform. It approximates total blood volume change per beat and can correlate with cardiac output.

Comparison

Table 2.3: PPG Dataset ML Model Comparison.

ML Model	R^2 score	MAE	MAPE	RMSE
Linear Regression	-1.9436	21.2316	18.21%	22.5357
Lightweight CNN	-0.5729	14.2101	11.75%	16.4736
XGBoost Regressor	-1.4461	19.0550	16.54%	20.5432
Support Vector Machine SVM)	-1.7733	20.0393	17.47%	21.8740
k Nearest Neighbour (kNN)	-2.1201	22.0185	19.01%	23.2015
ElasticNet	-1.4979	19.3175	16.46%	20.7598
ResNet34	-2.6995	22.7873	19.68%	25.2641
Transformer	-0.9013	15.8979	13.72%	18.1118
Hybrid CNN + Transformer	-0.0029	11.6569	10.15%	13.1543

Ranking (in terms of R^2 score)

1. Hybrid CNN + Transformer
2. Lightweight CNN
3. Transformer
4. XGBoost Regressor
5. ElasticNet
6. Support Vector Machine (SVM)
7. Linear Regression
8. k Nearest Neighbour (kNN)

On the GSR dataset, shown in Figure 2.6, the picture was more encouraging. The Transformer model achieved an R^2 of 0.7274 and an MAE of 12.02 mg/dL, outperforming the Hybrid CNN-Transformer (R^2 = 0.6772) and both lightweight alternatives. Crucially, the models that performed best on this dataset were those associated with Design Approach II, requiring server-side computation. The Lightweight CNN and Linear Regression, representative of on-device deployment, achieved substantially weaker results at R^2 values of 0.3160 and 0.1523 respectively, demonstrating that the accuracy gains available from deeper architectures are not merely incremental but qualitatively significant for this modality.

Figure 2.6: GSR Dataset (collected from Kaggle).

Comparison

Table 2.4: GSR Dataset ML Model Comparison.

ML Model	R^2 score	MAE	MAPE	RMSE
Linear Regression	0.1523	22.2954	20.42%	27.5690
Lightweight CNN	0.3160	18.8454	15.85%	24.7645
Transformer	0.7274	12.0232	11.07%	15.6337
Hybrid CNN + Transformer	0.6772	13.6722	12.19%	17.0113

Ranking (in terms of R^2 score)

1. Transformer
2. Hybrid CNN + Transformer
3. Lightweight CNN
4. Linear Regression

Taken together, these benchmarking results support two conclusions. First, the NIR and GSR combination is the most promising dual-sensor pairing, as NIR provides direct biochemical signals and GSR contributes complementary physiological information with meaningful standalone predictive capacity. Second, the models capable of leveraging this combined signal most effectively are deep architectures that cannot feasibly run on embedded hardware, pointing toward the server-based inference model of Design Approach II.

2.4.2 Monte Carlo Simulation Analysis

A Monte Carlo simulation was performed with 10,000 random glucose values from a normal distribution with a mean of 100 mg/dL and a standard deviation of 15 mg/dL to assess the statistical benefit of data diversity over data redundancy at the sensor level. Two scenarios were contrasted. In the redundancy case, typified by design approach I, the standard deviation of the noise level was the same for both sensors (5 mg/dL), and there was a high inter-sensor correlation (r = 0.8), suggesting that the errors of the two sensors were not independent but tended to move together. In the diversity case (Design Approach II), the two sensors were assumed to have different noise levels (5 and 7 mg/dL for the 2 sensors, respectively) and relatively low inter-sensor correlation (0.2), with the error of the two sensors being almost independent.

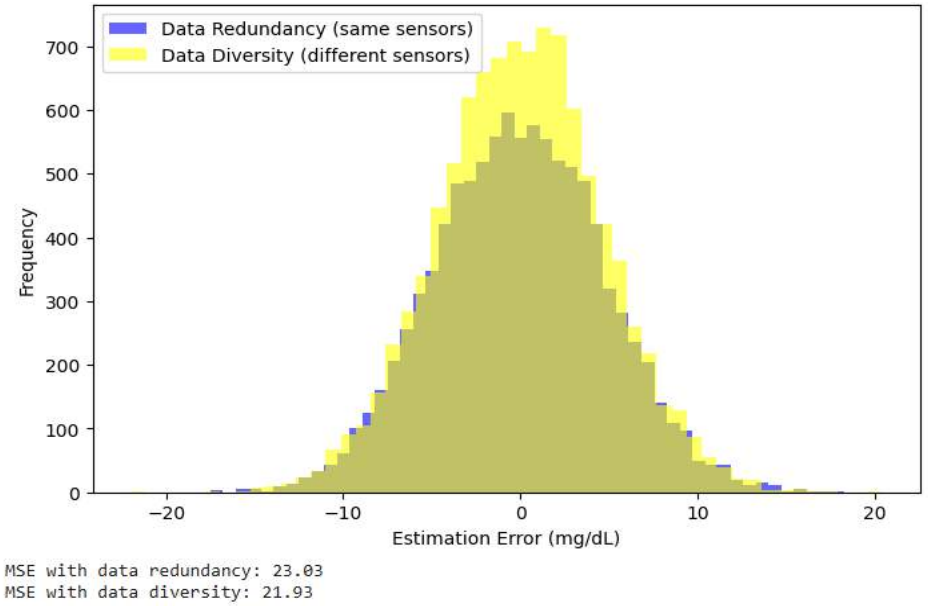


Figure 2.7: Results of Monte Carlo Simulation

The results, shown in Figure 2.7, indicated that the mean squared error (MSE) for the diversity configuration was lower (21.93) compared with the redundancy configuration

(23.03). The distribution of the errors under the diversity scenario was also a little tighter, and more symmetrically distributed around zero, suggesting fewer large errors in estimation. This is part of the ensemble estimation theory, which states that, if two estimators have uncorrelated errors, then the errors will partially cancel when they are averaged, even if one of the estimators is noisier than the other. The simulation thus gives a statistical argument for the dual-sensor diversity strategy, which is distinct from the results of the machine learning applied to the various modalities used above.

2.4.3 Cost and Power Analysis

The cost difference between the two approaches is an obvious benefit for Design Approach II. The on-board microprocessor needed for Design Approach I is estimated at BDT 16000 – 20000 alone, making the total cost of this approach to be BDT 26000 – 39000. The design approach II has a total project hardware cost of BDT 11,500 to 23,000 as it introduces the BLE communication module (BDT 1,000 to 1,500) on top of the lower cost microcontroller (BDT 11,000 to 21,500) and a minor web server setup cost. The revised sensor cost in the final optimal design budget of BDT 16,500-29,000 still leaves Design Approach II as a much more cost-effective configuration.

Table 2.5: Initial Budget of Design Approach 1.

Subsystem	Component Name and Specifications	Quantity	Cost (BDT)
Power	Battery Pack + Power Management	1	1000 - 1200
Sensing	Sensor	2	500 -1000
Analog Circuits	Resistors, Capacitors, and Op-Amps	-	1500 - 2000
Processing	Microprocessor	1	16000 - 20000
Display	LCD Display	1	500 - 800
PCB	Custom PCB (Sensor Interfacing)	1	3000 - 5000
Housing	3D-Printed Enclosure	1	1000 - 3000
Others	-	-	2000 - 5000
		Total	26000-39000

Table 2.6: Initial Budget of Design Approach 2.

Subsystem	Component Name and Specifications	Quantity	Cost (BDT)
Power	Battery Pack + Power Management	1	1000 - 1200
Sensing	Sensor	2	500 -1000

Analog Circuits	Resistors, Capacitors, and Op-Amps	-	1500 - 2000
Processing	Microcontroller	1	1000 - 1500
Communication	Communication Module	1	500 - 1500
Web Server Setup	Flask (Python)	-	0 - 1000
Display	LCD Display	1	500 - 800
PCB	Custom PCB (Sensor Interfacing)	1	3000 - 5000
Housing	3D-Printed Enclosure	1	1000 - 3000
Others	-	-	2000 - 5000
		Total	11500 - 23000

The power consumption is also a game-marker. Design Approach I distributes all the inference workload to the device, which translates to a total power consumption of 1,720 to 2,830 mW, mostly provided by the microprocessor. In Design Approach II, a server takes off the load for inference, bringing the total subsystems power consumption down to 460 to 790 mW. This reduction has direct consequences for battery life; a standard battery pack will last up to 6 to 8 hours (sufficient to meet the specification) with the power envelope of Design Approach II, but the power envelope of Design Approach I will significantly impact that target.

Table 2.7: Design Approach 1 Power Estimation

Subsystem	Estimated Power (mW)
Dual Sensor	80-120
Analog Circuit	40-60
Microprocessor	1500-2500
Display	100-150
Total	1720-2830

Table 2.8: Design Approach 2 Power Estimation

Subsystem	Estimated Power (mW)
Dual Sensor	120-80
Analog Circuit	40-60
Microcontroller	80-150
Wireless Communication	120-250
Display	100-150
Total	460-790

2.5 Conclusion

This chapter compares the different architectures to show that the non-invasive glucose monitoring device is optimally designed by combining two sensors with machine learning on the server. This final result comes from the convergence of three separate analytical streams. Through machine learning benchmarking on proxy datasets, it was found that the deep learning models used for meaningful accuracy in the GSR modality are too expensive to be computed on embedded hardware, while the combination of NIR and GSR sensor pairing has the best combined predictive foundation of the modality combinations explored. The Monte Carlo simulation verified that the data diversity approach between complementary sensors provided statistically less estimation error than the redundant dual-channel approach of Design Approach I, while cost and power analysis demonstrated that offloading inference to a server significantly lowers the cost and power consumption to be within the project's affordability and battery life constraints.

Importantly, Design Approach II also provides a longer term practical benefit that is not necessarily evident from purely performance-based metrics: the capability to centrally retrain and update the inference model as new project specific data are collected without any changes to the physical device. It is not a trivial feature, but an important one that allows this system to be flexibly improved with future developments of the system. The following chapters present how this chosen architecture has been optimized, modeled, and tested in individual hardware and software components. The complete table comparing the performance of the two design approaches is shown in Table 2.8.

Table 2.8: Design Approach Performance Comparison.

Performance Metric	Design Approach 1	Design Approach 2
Accuracy		✔
Cost		✔
Latency	✔	
Power Consumption		✔
Ease of Maintenance/Updates		✔

Chapter 3: Use of Modern Engineering and IT Tool. [CO9]

3.1 Introduction

The design of a non-invasive glucose monitoring system is a multidisciplinary engineering project, by definition, that involves analog hardware design, embedded firmware, wireless communication, and server-side machine learning inference. This is such a complex system and there is no one tool available to design, validate and optimize the system. Each of these subsystems works on a different physical principle and operates on a different time scale, and the effects of design mistakes in each of these subsystems are quite different: If a fault occurs in an analog front-end circuit, the result is different than if it occurs in a signal processing pipeline, or a wireless communication stack, or a machine learning model. It is hard to detect such errors manually while prototyping, as this is very time consuming and expensive in the early design phase and would give only partial information of the reasons for failure or the parameter space for optimisation.

The solution is provided by modern engineering and IT tools, which enable individual subsystems to be modelled, simulated and validated using software before committing hardware. The simulation first approach is more than just a matter of convenience—it is a methodology; the designer must be specific enough to the behaviour of the system to be able to program it into a simulation environment, bring to the surface assumptions that may not have been attended to, and produce quantitative performance data for comparison with prototype results. The tools used for this project were not just picked out of the air. Each was designed to address the specific aspects of the subsystem under investigation, and each has its own assumptions and limitations which need to be recognized and considered when interpreting results.

The chapter records the selection, justification and use of four engineering and IT tools used within the FYDP-D simulation phase: Proteus Design Suite, MATLAB Simulink, Google Colab and Cisco Packet Tracer. These tools combine to cover the entire system architecture, from analog signal conditioning and biosignal processing, all the way through to wireless communication feasibility and benchmarking of machine learning models. The chapter ends with a critical examination of the predictive value and limitations of each tool used in the context of the project.

3.2 Select Appropriate Engineering And It Tools

Selection of tools for a complex engineering project isn't as simple as finding the software which is technically able to perform a task. It involves matching the fidelity of the model used in the tool with the level of model detail available at the stage of development; making sure that the assumptions used by the tool do not systematically misrepresent the phenomena being simulated; and taking into account practical constraints such as tool's cost, accessibility, and learning curve for the team.

Four tools, shown in Table 3.1, were deemed to be suitable for the ongoing project phase and support different subsystems and/or different validation goals.

MATLAB Simulink was chosen for biosignal pipeline simulation due to its being the industry standard environment for block-diagram based modelling of continuous-time and discrete-time signal processing systems. It offers a library of pre-existing blocks for signal processing such as filters, feature extraction functions and data fusion operators, and it enables a complete processing pipeline of the multimodal data stream to be constructed and tested at the system level without having to develop low-level algorithms from scratch. The scope blocks in Simulink offer

real-time visualization of signal behavior at all stages of the pipeline, allowing you to quickly detect anomalies such as filtering anomalies, feature instabilities and fusion anomalies. In particular, Simulink models can be parameterised and swept over operating conditions, allowing for testing of pipeline robustness on a variety of simulated noise levels and signal amplitude corresponding to physiological variability in the real world.

Proteus Design Suite was used to simulate the analog front end circuit because it has the ability to simulate the specific circuit family of components employed in the GSR and NIR circuits, including the AD620 instrumentation amplifier, the TL081 operational amplifier, and the NE555 timer circuit used as a charge pump power supply. Schematics drawn with realistic models for the components can be simulated with Proteus to generate voltage and current waveforms at any node in the circuit. This ability is crucial in order to check that, prior to sourcing any physical components or soldering, each analog stage, the Wheatstone bridge, the TIA, the bandpass filter, the non-inverting amplifier, and the offset stage act as analytically predicted. It also enables the operating points to be checked against the datasheet of each component, and any potential problems, including op-amp output saturation or insufficient signal gain, can be identified early in the design cycle.

Cisco Packet Tracer was chosen to simulate wireless communication since it can simulate the behaviour of BLE devices such as advertisement, discovery, pairing and data exchange without any need for physical devices. If reliable low-latency BLE communication between the sensor device and connected laptop is a functional requirement, simulation prior to the availability of the hardware will alleviate the risk of discovering protocol-level incompatibilities during prototype testing. Simplified in comparison to a full RF propagation simulator, Packet Tracer's range and latency models are adequate to validate the basic communication viability for the 10-metre range that has been specified for the system.

Google Colab was chosen for machine learning model development and benchmarking for its ability to offer a Python notebook environment for the cloud, with access to GPU acceleration, and a full range of pre-installed scientific computing and deep learning libraries such as NumPy, Pandas, Scikit-Learn, TensorFlow, and PyTorch. By running machine learning experiments in Colab, you eliminate the need for high-performance local hardware, collaborate with others from around the world, and use the computing power to train and test architecturally complex models like Hybrid CNN-Transformer within a reasonable time frame. Colab also enabled a Monte Carlo simulation to be performed to compare redundant and diverse dual-sensor fusion strategies, based on Colab's ability to generate a large number of random numbers and visualisation libraries.

Table 3.1: Selection of Modern Engineering/IT Tools.

Engineering/IT Tool	Description	Why It Was Chosen
Google Colab	Google Colab is a cloud-based Python notebook environment that supports GPU/TPU acceleration and allows running ML code directly in the browser. It provides pre-installed scientific libraries such as NumPy, Pandas, Scikit-Learn, and TensorFlow.	• Ideal for running machine learning models without requiring a high-performance local computer. • Easy collaborative workflow for team members. • Useful for Monte Carlo simulations because of fast random-number generation and Python visualization libraries. • Free, accessible from anywhere, and requires no installation.

Proteus (Circuit Simulation)	Proteus Design Suite is a professional circuit simulation tool that supports analog and mixed-signal simulations. It allows schematic design, component behavior testing, and real-time signal visualization.	• Ideal for verifying NIR and GSR analog front-end circuits before hardware implementation. • Supports op-amp stages, filters, amplifiers, and sensor interfacing.
MATLAB Simulink	Simulink is a block-diagram simulation environment used to model complete system-level signal processing pipelines.	• Perfect for simulating the full biosignal pipeline, including filtering, feature extraction, and fusion. • Designed for testing algorithm behavior under different conditions. • Helps ensure the integrated system works before implementing firmware.
 Cisco Packet Tracer	Cisco Packet Tracer is a networking simulation software used to model communication protocols, wireless links, and device connections.	• Used to simulate Bluetooth communication between the SBC and laptop. • Allows testing data packets, latency, and reliability without needing actual hardware. • Helps validate communication feasibility for the proposed system.

3.3 Use Of Modern Engineering And It Tools

3.3.1 MATLAB Simulink: Biosignal Processing Pipeline Simulation

A full multimodal biosignal processing pipeline was implemented and tested in Simulink to model the processing of the two selected sensing modalities (NIR and GSR) from raw signal generation to feature extraction and fusion. The simulation was coded in MATLAB Simulink 2024b with two parallel processing chains and a fusion block.

The first, signal generation block of the NIR processing chain generates a synthetic signal composed of a fundamental cardiac-rate sinusoid and harmonic components that represent the spectral structure of a real NIR transmittance signal that is modulated by the pulsatile blood flow.

The measurement noise is then added with a low-frequency baseline drift component to simulate the combined effect of shot noise of the photodetector and slow tissue motion, which is achieved by a noise injection block. The output from the filter block is a noisy signal which goes through a bandpass filter block that is designed to reject the baseline drift and high frequency noise from the input, but leave the cardiac frequency components. The filtered signal is then passed to a feature extraction block, where three summary statistics are computed: a measure of the mean amplitude, the standard deviation, and an approximate estimate of the heart-rate based on a simple peak detection algorithm.

The GSR processing sequence is also structurally similar. An electrodermal activity signal is generated through a signal generation block (SGB) that creates a slowly varying skin conductance (SC) signal and incorporates spontaneous skin conductance response (SCR) events. A noise injection block is used to inject Gaussian noise and occasional spike artefacts that mimic motion induced signal contamination. These artefacts are suppressed by low-pass smoothing and moving-average filtering and the mean conductance and standard deviation are computed in a feature extraction block, as is an SCR count based on the number of rapid conductance rises above a set threshold.

The three-element feature vectors are sent to the fusion block, which normalises the features, applies a weighted linear combination of the normalised features, and calculates the composite stress-metabolic score—fused multimodal output. The result of the scope clearly shows that all seven of the fusion outputs (NIR mean, NIR standard deviation, NIR heart-rate estimate, GSR mean, GSR standard deviation, GSR SCR count, and the final fused score) vary consistently and within bounds across the time window of the simulation. This shows that the idea of feature extraction and fusion logic is self-consistent and without any numerical instabilities in the simulated operating conditions.

The main drawback of this simulation is that the signals are not generated experimentally but rather synthetically. Waveforms used are parameterised approximations of real NIR and GSR signals and are not fully capturing the full complexity of the physiological variance between individuals, the morphology of motion artefacts nor the specific noise floor of the hardware components. The simulation can therefore confirm structural logic of the pipeline and output stability of the fusion system, but cannot be certain of what the system will do on the actual participant data. This will be corrected in FYDP-C with the pipeline to be validated with signals acquired from the hardware prototype.

3.3.2 Proteus Design Suite: Analog Front-End Circuit Simulation

The system was. Tested using Proteus, which is a tool that helps design and simulate the complete analog front-end for both sensor channels, as well as the power supply subsystem. The team created three subcircuits. Made sure they worked well within the simulation environment.

The power supply subcircuit is built around a 12 V battery. This battery generates the four voltage rails required by the system: +12 V, +5 V, −12 V and GND. The 12 V battery directly powers the op-amp stages. A 7805 linear voltage regulator provides a +5 V digital rail for the ATmega32 microcontroller and the LCD display. The negative rail is generated using a 555-timer-based charge pump circuit. This circuit produces −12 V, which is necessary to bias the negative supply pins of the operational amplifiers. The Proteus simulation confirmed that all four rails settle to their target voltages and remain stable across the simulated load conditions of the circuit. The power supply subsystem is a part of the system and the Proteus simulation helped test it.

The NIR subcircuit is designed to convert the photocurrent produced by the NIR photodiode into a voltage signal. The team used a calibrated source to inject a precise stable input current representing the expected photodiode output under typical illumination conditions. The first 741 op-amp stage is configured as a transimpedance amplifier. This amplifier converts the input to a proportional voltage. A parallel capacitor stabilises the feedback loop. Limits high-frequency noise. The subsequent RC network forms a low-pass filter to suppress residual high-frequency components. A second 741 gain stage amplifies the filtered voltage

to a level within the microcontroller's ADC input range. The NIR subcircuit is a part of the system and the Proteus simulation confirmed that it works correctly.

The GSR subcircuit is the complex of the three consisting of five cascaded stages. The Wheatstone bridge detection stage produces an output voltage proportional to the deviation of skin resistance from the bridge balance point. This low voltage is then applied to the inputs of the AD620 instrumentation amplifier. The bandpass filter stage isolates the relevant 0.5 to 5 Hz band of skin conductance variation. The non-inverting amplifier stage provides a gain bringing the signal to an amplitude suitable for the ADC. The level-shifting offset stage adds a DC component via a potentiometer to shift the amplified signal into the unipolar 0 to 5 V range required for digitisation. The GSR subcircuit is a component of the system and the Proteus simulation verified that it works as expected.

One limitation of the Proteus simulation is that it uses component models. These models do not fully represent the noise behaviour, temperature sensitivity and component-to-component variation of parts. The 741 op-amp model used in the NIR subcircuit does not capture the gain-bandwidth product that may affect circuit behaviour at the upper end of the signal frequency range, in a physical implementation. These discrepancies will become apparent during breadboard prototype testing, at which point component values may require adjustment. The system was. Tested using Proteus and the Proteus simulation helped identify potential issues with the system.

3.3.3 Cisco Packet Tracer: Wireless Communication Simulation

We used Cisco Packet Tracer to see if Bluetooth Low Energy would work between the sensor device, which's like a small computer and the laptop that sends information to the server. The sensor device was set up to send out signals. The laptop was set up to look for these signals.

The test went through the steps to set up a Bluetooth Low Energy connection. The sensor device sent out signals every 10 seconds. Had a special code to identify it. The laptop looked for these signals. Connected to the sensor device when it found them. Once they were connected we checked that they could send and receive information to each other. The test showed that Bluetooth Low Energy worked well when the devices were close to each other like in a room.

The results showed that the laptop found the sensor devices signals they connected without any problems. They could send information to each other. This means that Bluetooth Low Energy is a way for the sensor device and laptop to talk to each other.

There is a limit to how well Cisco Packet Tracer can simulate the real world. It does not take into account things like signals bouncing off walls or other devices interfering with the connection. So the 10-metre range we saw in the test is probably the best case scenario. In the world, with walls and other devices around the range might be shorter. We will test this again with a prototype to see how it works.

3.3.4 Google Colab: Machine Learning Benchmarking and Monte Carlo Simulation

Google Colab was used as the main computation environment both for the machine learning models benchmarking and the Monte Carlo sensor fusion simulation. The GPU-accelerated cloud infrastructure allowed to train and test computationally intensive architectures such as the Hybrid CNN-Transformer and the Transformer in reasonable time-frame that would not have been feasible with an ordinary PC.

The three proxy data-sets utilized during the ML benchmarking were the Kaggle NIR data-set which is a set of pairs of infrared sensor voltage value and reference glucometer reading of 26 subjects; the Mendeley PPG data-set that is a set of physiological signal features from 67 subjects with invasive blood glucose reading labels; and the Kaggle GlucoBench data-set, that contains 15731 entries and 31 features such as continuous glucose measurements and galvanic skin response readings. All three data-sets were chosen as the closest publicly available proxies for the modalities and the feature shapes that characterize the project's own data set and used for benchmarking models of a large variety from linear regression to deep learning ensemble models.

Each data set was pre-processed by imputation of missing values, normalization using Min-Max scaling and 80/20 split for training and testing using stratification. Each model was tested on the held-out set and the following measures were recorded: R value, Mean Absolute Error, Mean Absolute Percentage Error and Root Mean Squared Error. Results of the NIR dataset confirmed that lighter models yield satisfactory results on single modality low dimensional data-sets. The results of the PPG dataset demonstrate that the chosen architectures were unable to extract any reliable glucose-predictive signal from PPG features and this supports the exclusion of PPG from the chosen sensor-pairing. Deep learning models were found to greatly outperform the lightweight architectures on the GSR data-set: the Transformer achieved an R value of 0.7274, with MAE of 12.02 mg/dL, while the Hybrid CNN-Transformer achieved an R value of 0.6772 and an MAE of 13.67 mg/dL, both far outperforming the Lightweight CNN whose R value was 0.3160. The obtained results provide statistical support for the fact that it is only possible to leverage the accuracy potential of the GSR modality on the server through deep learning and thus strongly support the adoption of Design Approach II.

The Monte Carlo simulation (which also ran in Colab) generated 10000 synthetic glucose values according to a normal distribution and created two dual-sensor fusion scenarios: data redundancy, where correlation was high between both sensors (0.8) and noise levels were equal between both sensors, and data diversity, where correlation was low between both sensors (0.2) and noise levels were different for both sensors. Error mean squared value was higher for redundancy scenario (23.03) compared to diversity scenario (21.93) and the recorded error distribution was centered more symmetrically on a narrower peak. These results statistically prove that data diversity enhances sensor fusion accuracy and therefore support the adoption of the dual sensor-pairing in Design Approach II.
It must be mentioned that the data sets used in the benchmarking are only proxies of the project's data and are far from representative as the sensor readings on those data sets are not captured on the same instant for the given pair of sensors, this means that interactions and joint distributions will not be represented in the analysis of the proxy-data set and the obtained rankings may be biased. Those results shall therefore be interpreted as indications of the capability of the proposed architectures and not as precise prediction for the performance of the final system on the project-data set. Retraining the models and re-testing using project data will be performed in FYDP-C.

3.4 Conclusion

The four engineering and IT tools selected to aid in simulating the FYDP-D design cover the entire range of the proposed architecture from analog sensor front end, to the biosignal processing pipeline, and radio link, to the machine learning inference layer. Each tool was selected based on its aptitude in handling the modelling problem presented and each provides concrete, quantitative information to guide the design.

MATLAB Simulink validated that the multimodal feature extraction and fusion pipeline is structured correctly and that it remains numerically stable across the simulation environment. Proteus tested each analog subcircuit's operating point, and validated that signal amplitudes remain within range for each subcircuit such that they remain within the appropriate limits of the ADC input range across the expected range of physiology. Cisco Packet Tracer tested the feasibility of an over-the-air connection from the sensor to the relay laptop via the use of BLE technology across the system range. The Monte Carlo simulation of the ML layer's performance and the system's accuracy (as defined by the ML model's accuracy) both provided real quantitative data used to select Design Approach II. Perhaps even more critically, it is important to mention what cannot be done with these tools.

The models themselves that exist for all four software tools are simply simplifications of physical reality. The idealisation of components within Proteus, the creation of a signal stream using pure algorithms within Simulink, the abstract modelling of radio propagation within Packet Tracer, and the utilization of a substitute dataset within Colab each insert a degree of separation between the simulated output and physical behavior. This step therefore proves that the system can work based on a variety of models and test cases. The proof of concept will be demonstrated in hardware in the next step using hardware experiments and a custom dataset; here we establish an important benchmark upon which to judge those results.

Chapter 4: Optimization of Multiple Design and Finding the Optimal Solution. [CO7]

4.1 Introduction

Identifying a preferred design approach through comparative analysis, as carried out in Chapter 2, is a necessary but not sufficient step in the engineering design process. A design approach is, at that stage, still an architectural concept: a set of choices about sensor strategy, computational paradigm, and system topology that has been shown to be preferable to its alternatives on theoretical and simulation-based grounds. Before that concept can be realised as a functioning prototype, it must be subjected to a further layer of engineering work, one concerned not with which architecture to adopt but with how to implement that architecture as effectively as possible within the constraints of the project. That is the work of optimization.

In this context optimisation is not a mathematical process of tuning the parameters, although this is part of a process. More generally, the process of progressively refining each subsystem of the chosen architecture, the process of verifying the refinement by simulation, and the process of removing design uncertainties before the hardware is irrevocably committed. This is a daunting task for a system as technically diverse as the one proposed here, where multiple levels of abstraction need to be worked simultaneously: the circuit level, where each analog conditioning stage has to operate within its specified operating envelope; the signal processing level, where the feature extraction and fusion pipeline has to produce stable, discriminative outputs under realistic noise conditions; the communication level, where the wireless data link must act reliably within its specified operating range; and the machine learning level, where the model architecture has to maximise predictive accuracy for the available proxy data while still being computationally viable within the server-side deployment context.

The optimization process in the four subsystems of Design Approach II is documented in this chapter. In section 4.2, the engineering decisions that were made to improve the chosen design are described, providing in-depth reference to the evidence gained from the simulations used to guide the decisions taken. The results of optimisation are summarised in Section 4.3, which is a formal statement of the optimum design solution. The developed solution is subjected to a structured performance evaluation in Section 4.4 and compared to the requirements and success criteria given for the project. Section 4.5 draws conclusions and sets the stage for the hardware prototype work that will follow in FYDP-C.

4.2 Optimization Of Multiple Design Approach

4.2.1 Analog Front-End Circuit Optimisation

The analog front-end represents the point at which the physical world, specifically the body's optical and electrodermal responses, is converted into digital data that the microcontroller and downstream software can process. Errors introduced at this stage propagate irreversibly through every subsequent layer of the system; no amount of signal processing or machine learning sophistication can recover information that was lost or distorted during acquisition. Analog front-end optimisation therefore received priority attention.

4.2.1.1 NIR Signal Conditioning Chain

Tiny light signals from the near-infrared diode create minute electric currents - just millionths of an amp. Because the ATmega32 chip needs voltages between zero and five to measure anything, those faint pulses must become stronger readable levels. Background interference could hide the useful data tied to glucose changes, so cleaning up unwanted disturbances matters just as much. Instead of stacking parts without testing, each choice in shaping the circuit was checked first using Proteus software, shown in Figure 4.1. What finally went in matched what the model predicted under real-like conditions. Right away, the setup picked a transimpedance amplifier layout for turning tiny currents into voltage, built around a 741 chip fitted with a 500 thousand ohm resistor and a 100 nanofarad cap wired side by side. Rather than tossing in just a single resistor, this approach keeps the diode's connection point near zero volts, sidestepping issues caused by internal charge storage that could warp response at certain tones or spark wild swings. That added capacitor? It tunes the circuit's reach slightly past the highest heartbeat-linked frequencies so unwanted hiss fades out while pulse signals stay clear. Tests ran inside Proteus showed steady behavior no matter how weak or strong the incoming flow got, plus all resulting voltages stayed safely where the amplifier works best.

Next came a basic RC low-pass setup placed right after the TIA output, ahead of the next amplifier step. Built into the path, it quietly cuts very fast noise using almost no extra parts. It avoids the timing issues an active version could bring if stuck there. Slowness of the resistor-capacitor pair matches how fast signals can pass through what comes next. That slowness sets just where higher tones drop out. Next came a second 741 op-amp setup wired for non-inverting voltage boost - this lifted the cleaned-up TIA signal into line with what the ADC could handle. That stage's multiplication factor landed so peak signals hit about four-fifths of max ADC capacity, leaving room when spikes pop up unexpectedly. Testing the entire near-infrared path inside Proteus showed a sample 100 nanoampere input made the final voltage land right where it should - for solid proof both gain steps were sized correctly.

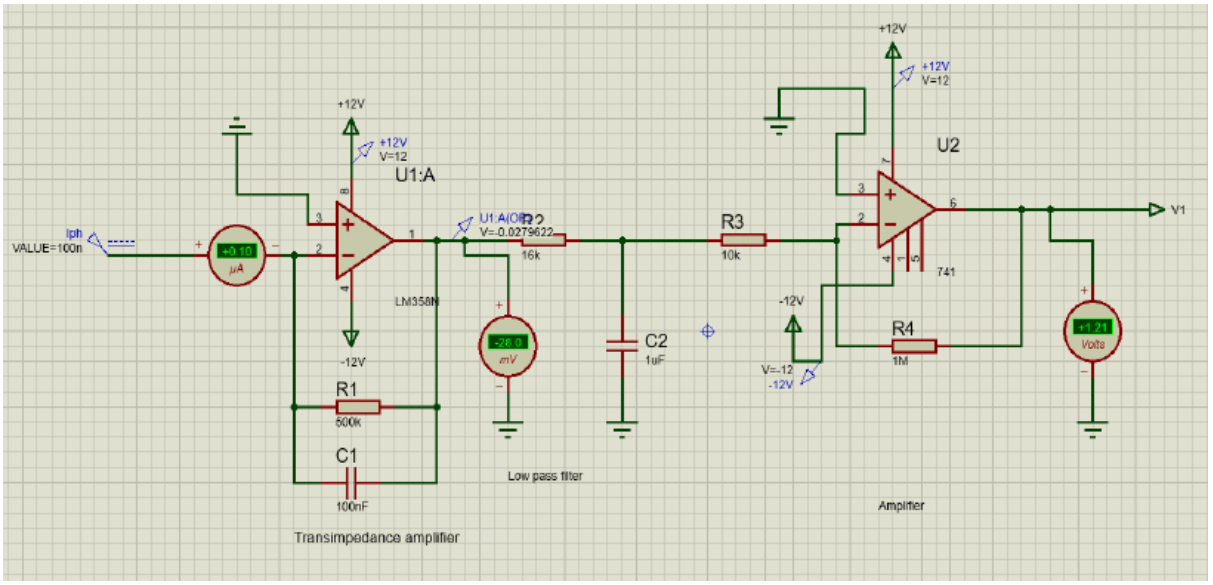


Figure 4.1: NIR Subcircuit.

4.2.1.2 GSR Signal Conditioning Chain

The GSR front end is also significantly more complex than the NIR chain–five cascaded stages that will amplify the tiny differential voltage output of the Wheatstone bridge into a clean, unipolar voltage ready for digitisation. All stages were optimized, one by one, and then tested due to one another.

This Wheatstone bridge detection stage was biased by using a precision voltage divider, so that it was centered at 0.5 V. The applied voltage was used only below approximately 0.5 V and the bias was not chosen at random as the skin conductance will only be linear to voltage below this level, and above this level, an electrochemical reaction at the electrode-skin interface will occur which will bias the measurement. The bridge designed to give the required differential output range with a millivolt and submillivolt output specification was matched arms 2 kΩ with a 47 kΩ matching resistor, so that the resistive change to be measured under a normal physiological state would lie within the linear range of the instrumentation amplifier output. On the basis of three properties that are of special importance in this application, the pre-amplification stage of the measurement was realized using the AD620 instrumentation amplifier. It has high input impedance (about 10 GΩ), thus doesn't load the bridge and cause the Bridge to unbalance, it has a differential input as the common-mode noise that affects the bridge output voltage is rejected, the gain can be set with a single outside resistor, which avoids the difference in gain of the two resistors in a conventional op-amp circuit. It was programmed for a gain that would ensure adequate amplification was achieved to enable the subsequent filter stage to handle the millivolt bridge output level.

An active bandpass filter constructed using cascaded stages of TL081 op-amps was set to a low pass cutoff of 5 Hz and a high pass of 0.5 Hz, thus isolating the frequency band (0.5-5 Hz) equivalent to the physiologically relevant frequency band of the skin conductance response (SCR). This application chose the TL081 over the more well-known UA741 due to its lower noise, lower input offset voltage and cost-effectiveness. The low-pass section provides a drop-off of frequencies above 5 Hz, such as 50 Hz powerline interference and motion-induced muscle noise. The high-pass section removes frequencies below 0.5 Hz which include polarization effects of the electrodes and slow baseline drift. The predicted attenuation properties at both the low and high energizations were verified by Proteus simulation. The second stage is a non-inverting amplifier circuit also made with the TL081 chip with a 100kΩ feedback resistor and a 1kΩ reference resistor, giving an amplification of about 100. The gain level was chosen to (somewhat) align the filtered GSR signal amplitude with ADC input range while not exceeding maximum measured skin conductance excursion.

The last obstacle to digitisation was the amplified GSR signal, which has the potential for a positive or negative voltage range; however, the ATmega32 ADC only accepts positive voltage. An offset stage is used to remove any voltage shift level from the amplified GSR signal, which clarifies the bipolar nature of the signal. A unity gain adder circuit adds the amplified GSR signal to an adjustable DC offset (positive output) from a potentiometer that can be referenced externally to the positive supply rail. The potentiometer can be adjusted to

bring the signal to a point around 2,5 V which will cause the bipolar excursions to be centered in the unipolar window of the ADC, avoiding clipped excursions, thus tuning the operating point or set-point of the signal. The full 0-5 V output range was confirmed there over the expected range of conductances using the Proteus simulation of the entire five stage GSR chain for representative physiological inputs. The explanation of the design is performed from the start but step by step:

- Detection of the Skin's Resistance through Wheatstone Bridge

The Wheatstone bridge is used here because it's a highly sensitive and accurate way to measure small resistances (like the resistance of skin). It balances the voltage across four resistors and outputs a small voltage difference that's proportional to the resistance change in one of the arms of the bridge and in this case, it is the skin's resistance. This is crucial because the change in skin resistance due to sweating is very small. Here, this bridge gives a highly accurate and stable reading of the skin's resistance, which is essential for precise measurements.

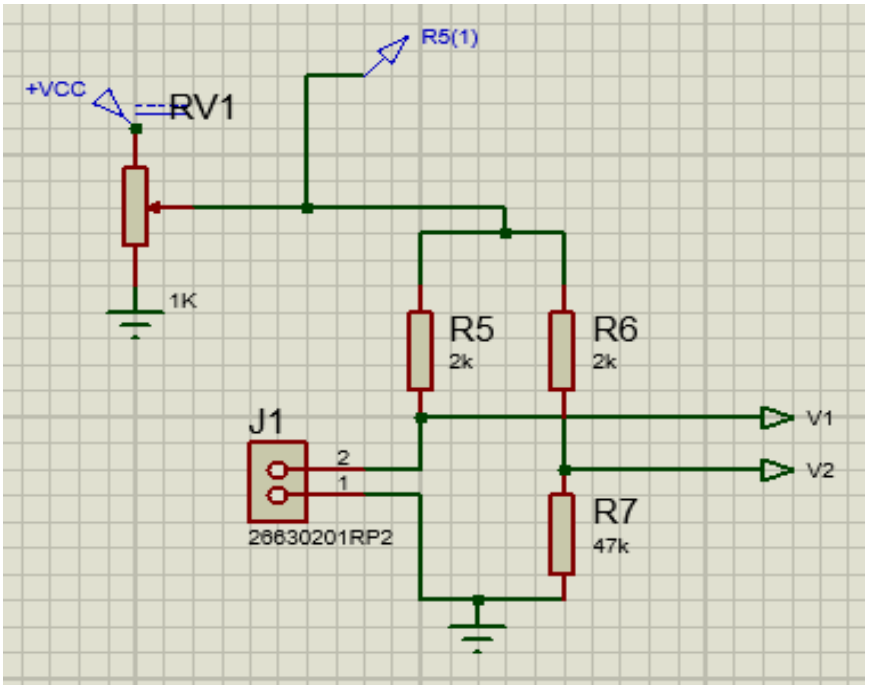


Figure 4.2: Detection through wheatstone bridge

The detection stage of the GSR circuit utilizes a Wheatstone bridge that is supplied with a fixed 0.5V through a voltage divider (as shown in Fig). This low voltage is essential because skin conductivity only exhibits a linear relationship with the applied voltage when it is below 0.5V.

The Wheatstone bridge configuration consists of four resistors, with the skin's resistance being one of them. The voltage across the bridge, denoted as V1 and V2, is used to calculate the skin conductivity. The skin's resistance can then be extrapolated by linking the two ends of the bridge to a pre-amplifier.

A significant point to note is that, due to the high input impedance of the AD620 instrumentation amplifier used in the design, impedance matching is not necessary. The AD620 provides a high input impedance of approximately 10 GΩ with a capacitance of 2 pF, which is more than sufficient to ensure accurate measurements without the need for additional voltage followers.

Here, Wheatstone Bridge Output:

The voltage output from the Wheatstone bridge can be expressed as:

$$V_{out} = V_{in} \times \left(\frac{R_2}{R_1+R_2} - \frac{R_4}{R_3+R_4}\right) \quad \text{(Eq. 4.1)}$$

Where:

- Vin is the applied input voltage (0.5V),
- R1,R2,R3,R4 are the resistors in the Wheatstone bridge, with R2 and R4 representing the skin's resistance and the other resistors providing balance.

The conductivity (σ) of the skin can be calculated as:

$$\sigma = \frac{1}{R_{skin}} \quad \text{(Eq. 4.2)}$$

where R_{skin} is the skin's resistance obtained from the bridge above.

- Pre-amplification Stage (Using AD620 Instrumentation Amplifier)

The signal from the Wheatstone bridge is very weak (measured in millivolts), so it needs to be amplified without losing its integrity. The AD620 instrumentation amplifier is specifically chosen because it offers high input impedance (which ensures that it does not load the Wheatstone bridge and thus distort the signal) and low noise (important when measuring small signals). It amplifies the small GSR signal to a level that can be further processed in the next stages.

In our design the R4 (RG - responsible for gain) resistor programs the AD620 gain, or more accurately, any impedance that emerges between Pins 1 and 8. With resistors ranging from 0.1% to 1%, the AD620 is made to provide precise gains. The necessary RG values for different gains are displayed in the table below. For G = 1, the R8 pins are unconnected (R8 = ∞).

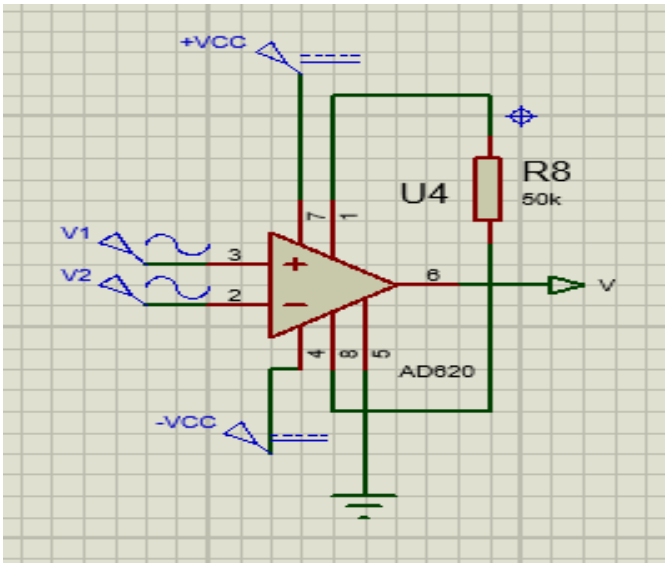


Figure 4.3: Pre-amplification with AD620.

RG (R8) can be computed using the following method for any arbitrary gain:

$$G = \frac{49.9\ k\Omega}{R_G} + 1 = 2 \text{ [Since, } R_G \text{ is considered to be 49.9 in the first row]}$$

Table 4.1: Effect of Resistor Tolerance on Gain

1% Std Table Value of RG (Ω)	Calculated Gain	0.1% Std Table Value of RG (Ω)	Calculated Gain
49.9 kΩ	1.990	49.3 kΩ	2.002
12.4 kΩ	4.984	12.4 kΩ	4.984
5.49 kΩ	9.998	5.49 kΩ	9.998
2.61 kΩ	19.93	2.61 kΩ	19.93
1.00 kΩ	50.40	1.01 kΩ	49.91
499 Ω	100.0	499 Ω	100.0
249 Ω	199.4	249 Ω	199.4
100 Ω	495.0	98.8 Ω	501.0
49.9 Ω	991.0	49.3 Ω	1,003.0

- Filter Stage (Band-Pass Filter)

The GSR signal can be contaminated with noise, especially from power lines (50 Hz) or motion artifacts (such as from muscle activity). A band-pass filter is designed to remove high-frequency noise (such as 50Hz power line interference) and low-frequency noise (like muscle or movement artifacts), while allowing the relevant GSR signal frequencies to pass through. The 0.5-5Hz range is typically the range of skin conductance changes that reflect emotional arousal, so the filter ensures that only this important signal is processed.

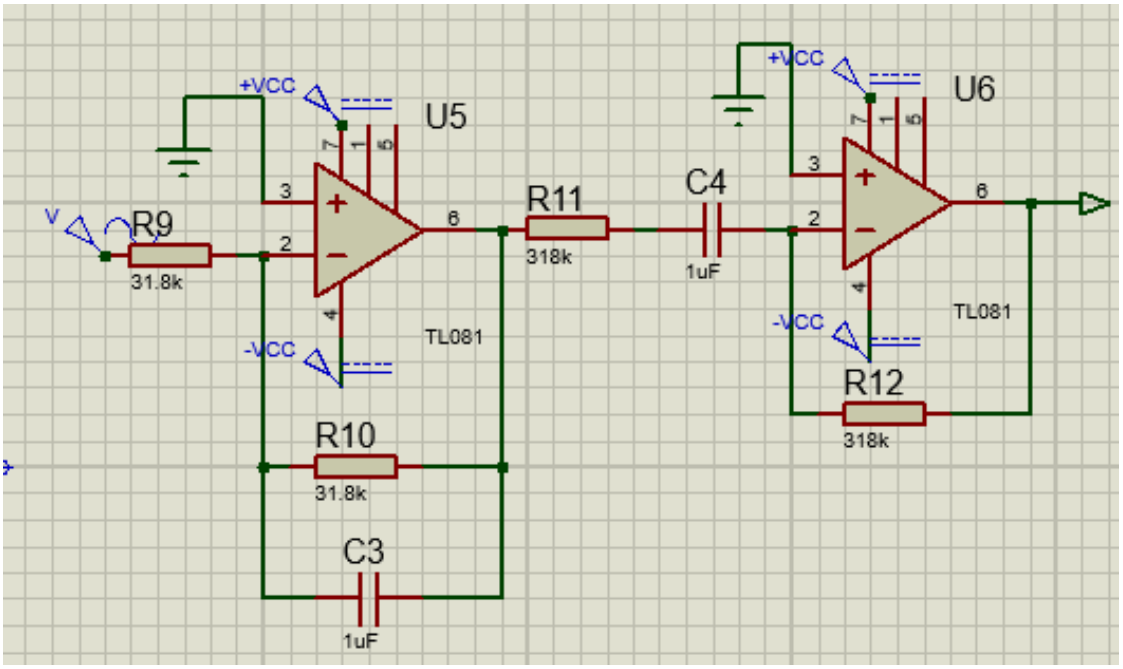


Figure 4.4: Filter stage (band-pass filter)

In this design, we have chosen an active band-pass filter. The filter is created by cascading two filters:

1. Low-pass filter with a cutoff frequency (fc1 = 5 Hz).
2. High-pass filter with a cutoff frequency (fc2 = 0.5 Hz).

These filters are implemented using the TL081 operational amplifier, chosen for its low noise, low input offset error, cost-effectiveness, and availability compared to other general-purpose op-amps like the UA741 or TL071.

Low-Pass Filter:

The low-pass filter is designed to pass signals with frequencies below the cutoff frequency (fc1 = 5 Hz) while attenuating higher frequencies. The cutoff frequency is given by the following equation:

$$f_{c1} = 1/2\pi R_{10} C_3 \quad \text{.....(Eq. 4.3)}$$

Where,

R10 and C3 are the resistor and capacitor values in the low-pass filter.

High-Pass Filter:

The high-pass filter is designed to pass signals with frequencies above the cutoff frequency (fc2 = 0.5 Hz) and attenuate lower frequencies. The cutoff frequency is given by:

$$f_{c2} = 1/2\pi R_{12} C_4 \quad \text{.....(Eq. 4.4)}$$

Where,

R12 and C4 are the resistor and capacitor values in the high-pass filter.

- Amplification Stage (Non-Inverting Amplifier)

After filtering, the signal is still too weak to be processed by digital systems like an Arduino (which typically accepts 0-5V signals). The non-inverting amplifier stage provides a controlled gain to the signal, ensuring it reaches an adequate level for further processing.
The gain must be chosen carefully to avoid clipping (which could distort the signal) and to ensure the signal stays within the range of the Arduino's ADC (analog-to-digital converter).

Here, we used a non-inverting amplifier TL081 with the following gain formula:

$$G = 1 + R2/R1 \quad \text{.....(Eq. 4.5)}$$

For this case, G = 1 + R13/R14

= 1 + 100/1

= 101 (can be considered as 100)

This gain is chosen such that the GSR signal must not exceed 5 V maximum (corresponding to the maximum voltage of the Arduino board input) under any physiological conditions.

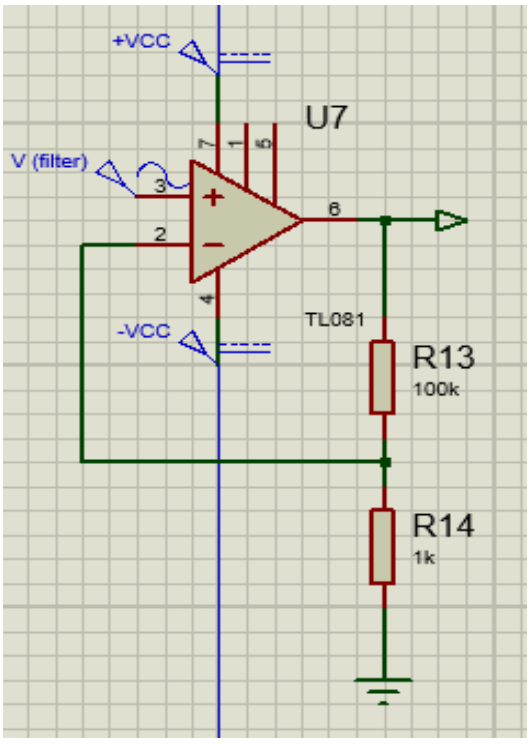


Figure 4.5: Amplification (non-inverting)

- Offset Stage (Level Shifting)

The output from the previous amplification stage might have both positive and negative voltage swings. However, the Arduino ADC can only handle positive voltages (0-5V). This stage adds an offset to the signal to shift it into the proper range (typically 0 to 5V) for the Arduino's ADC, ensuring it can be correctly digitized and processed.

The GSR signal is bipolar after amplification. However, an offset circuit is required because the Arduino acquisition board's analog input is unipolar (must range between 0 and 5V maximum).

This circuit is an operational amplifier set up as an adder (with a gain of 1) that allows the GSR signal to be shifted towards the positive portion by adding a DC component. The voltage (+ Vcc) and the potentiometer RV2 are used to create the DC component. Through R16 and R15, respectively, this voltage and the GSR voltage are delivered to the input (+) with identical values of 1KΩ.

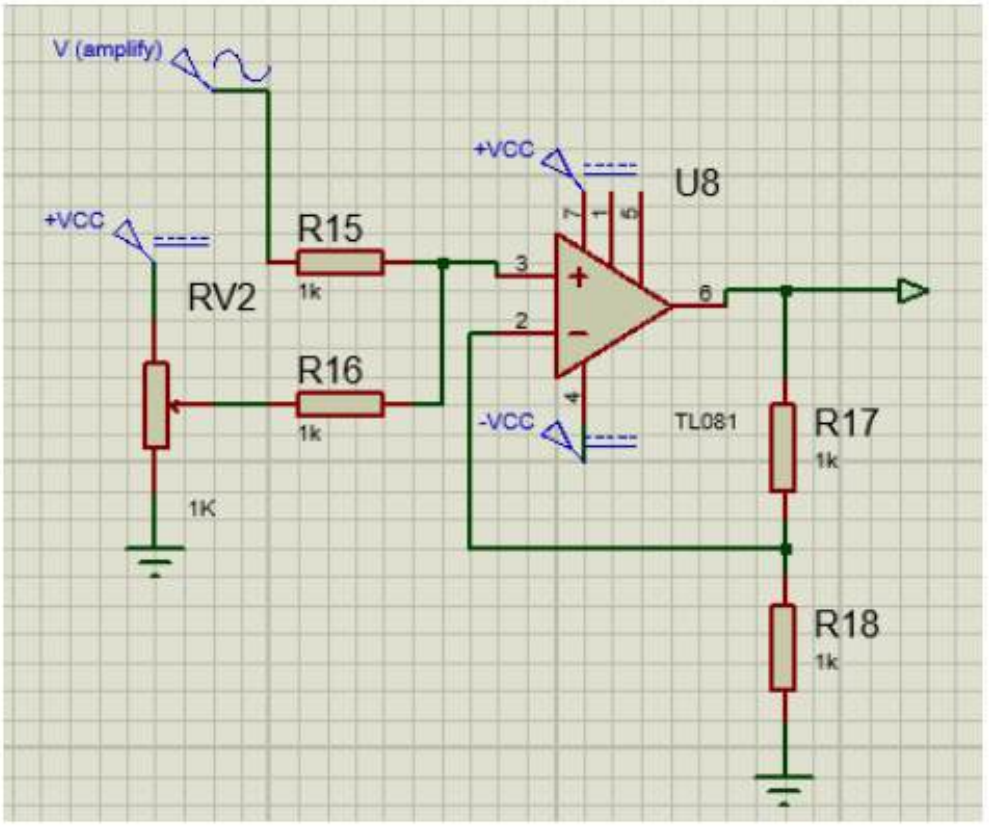


Figure 4.6: Offset Stage

4.2.1.3 Power Sub Circuit

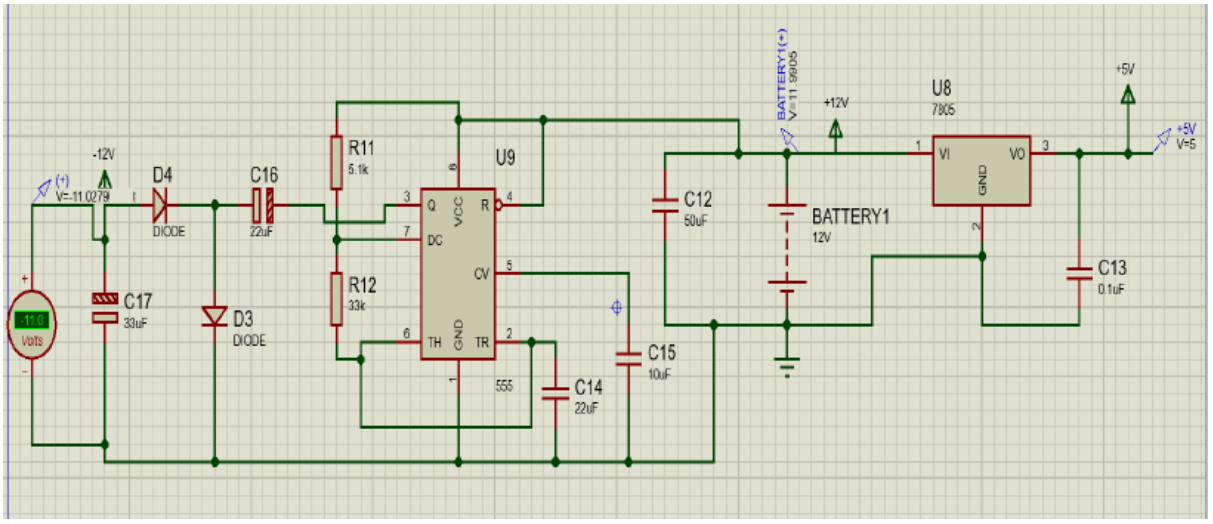


Figure 4.7: Power Supply Circuit.

The power circuit is built around a single 12 V battery and generates four rails required by the system: +5 V, +12 V, −12 V and GND. The raw 12 V from the battery directly serves as the +12 V analog rail, which powers the op-amps and analog conditioning blocks. A 7805 linear regulator is connected to the battery through decoupling capacitors to provide a stable +5 V digital rail for the ATmega32 and LCD, improving noise immunity and ensuring that the logic devices operate within their specified supply limits. To obtain the negative rail, the design uses a charge-pump circuit driven by a timer/oscillator, 555 alternately charges and inverts the supply, producing approximately −12 V referenced to ground, which then biases the negative supply pins of the op-amps so they can handle bipolar signals around 0 V.

4.2.2 Biosignal Processing Pipeline Optimisation

With the analog front-end validated at the circuit level, the second optimization domain concerns the digital signal processing operations applied to the digitised sensor outputs before feature vectors are assembled for transmission to the inference server. This work was carried out in MATLAB Simulink 2024b, shown in Figure 4.8.

The NIR processing chain in Simulink models the signal after ADC digitisation. A bandpass filter block isolates the cardiac-rate frequency components of the digitised NIR signal, suppressing the residual baseline drift and high-frequency noise that passed through the analog filter stages. The feature extraction block then computes three summary statistics: the mean amplitude of the filtered signal over the acquisition window, the standard deviation of that signal as a measure of beat-to-beat variability, and an approximate instantaneous heart rate derived from the intervals between detected systolic peaks. These three features were selected on the basis of their established physiological relevance to perfusion and vascular tone, which are properties that modulate the optical path length through tissue and therefore carry indirect glucose-relevant information.

The GSR processing chain applies a low-pass smoothing filter and a moving-average filter to the digitised conductance signal, further suppressing the high-frequency noise components that passed through the analog bandpass filter. The feature extraction block computes the mean conductance level, the standard deviation of conductance over the acquisition window, and a count of skin conductance response events, defined as rapid conductance rises exceeding a defined threshold amplitude within a minimum inter-event interval. The SCR count is particularly informative as a feature because it reflects the rate of sympathetic nervous system activation, which is known to correlate with metabolic arousal states associated with glycaemic variability.

The fusion block normalises the six features extracted from both chains to the same scale, performs a weighted linear combination to create a composite physiological state score, and combines all the normalised features in a single vector for transmission. The weights for the linear combination were heuristically determined for the simulated scenario to represent the relative signal quality of each modality in the simulated scenario, while in a machine learning model they will be automatically learned from the data. Simulink scope outputs, shown in Figure 4.9, confirmed that all seven fusion outputs evolve stably across the simulation window without numerical overflow, saturation, or oscillatory artefact, validating the pipeline's structural integrity under the simulated operating conditions.

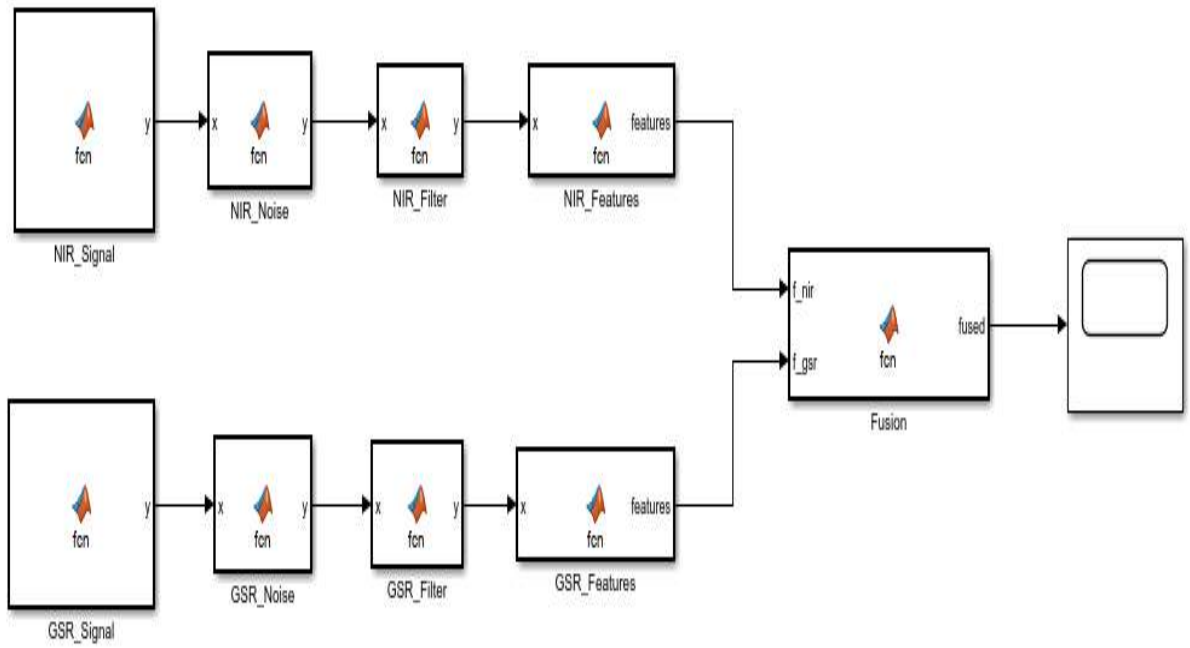


Figure 4.8: Simulink Block Diagram

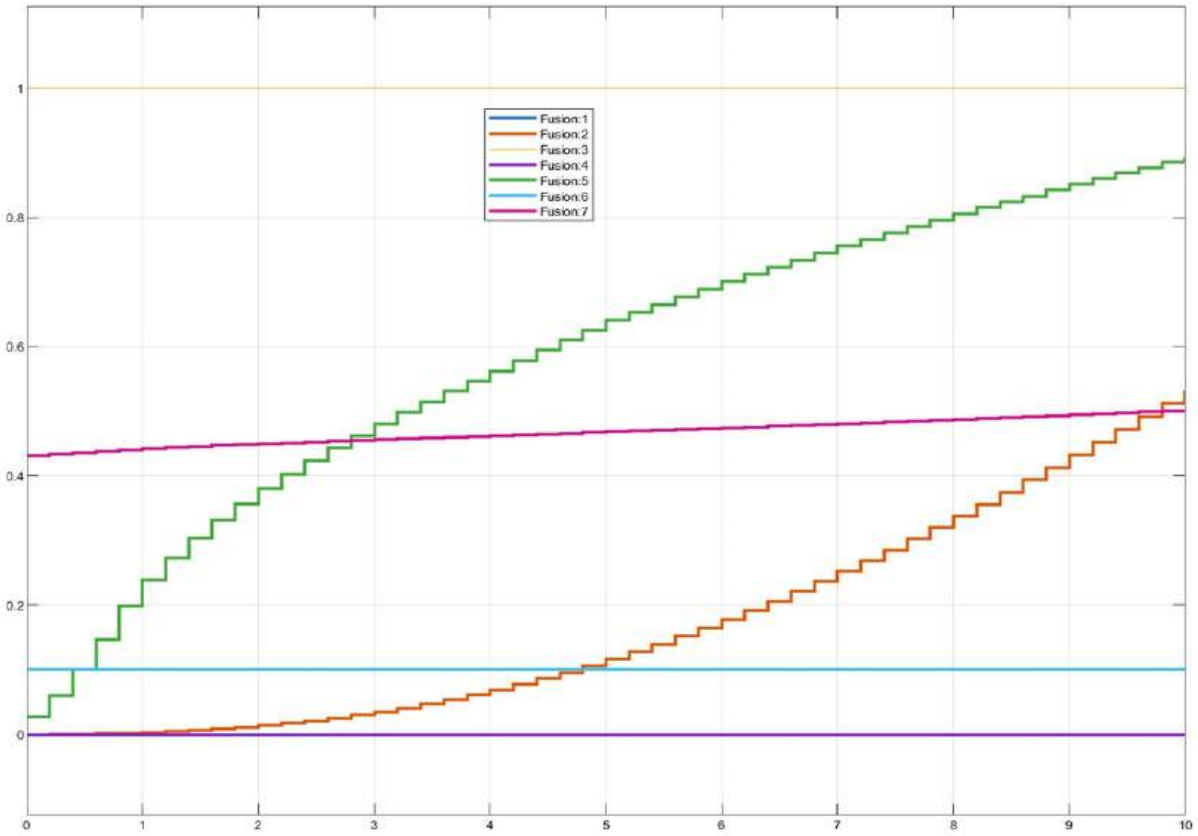


Figure 4.9: Simulink Output

4.2.3 Wireless Communication Optimisation

A test inside Cisco Packet Tracer checked how well the Bluetooth Low Energy link works after changes were made. Because reliability matters most, each setting - like how often signals are sent, which IDs get assigned, and when connections form - was put under review. Even at the full ten metres the gadget is meant to cover, messages must arrive without delay or loss. With those values locked in, repeated runs showed steady performance across trials. No guesswork remained once every outcome matched what the design needed.

Starting every 10 seconds, the SBC sent out signals using a distinct ID so it wouldn't mix up with nearby Bluetooth gadgets. On the receiving side, the laptop stayed alert, constantly searching for those signals like a radio tuned to one station. When the test ran, each message from the small computer showed up clearly on screen, no hiccups. Connection sparked fast, almost instantly, once contact happened between both sides. Data moved back and forth smoothly - reading values here, updating them there - with nothing stuck or lost along the way. Each step played out exactly as expected, repeat after repeat, during the whole run.

Something clear came up in the simulation about where to place the devices. As gadgets move closer to the edge of the 10-metre limit, their BLE signal drops off noticeably - distance matters more here. When measuring, though, the sensor and laptop will usually sit just one or two meters apart, so that drop-off won't cause issues. Pulling everything together - the setup from Chapter 2 and what showed up in testing - it holds true: BLE works well enough between sensor and relay, and the settings used fit fine with how fast and reliably the system needs to run every day.

Figure 4.10: Initial Setup in Packet Tracer.

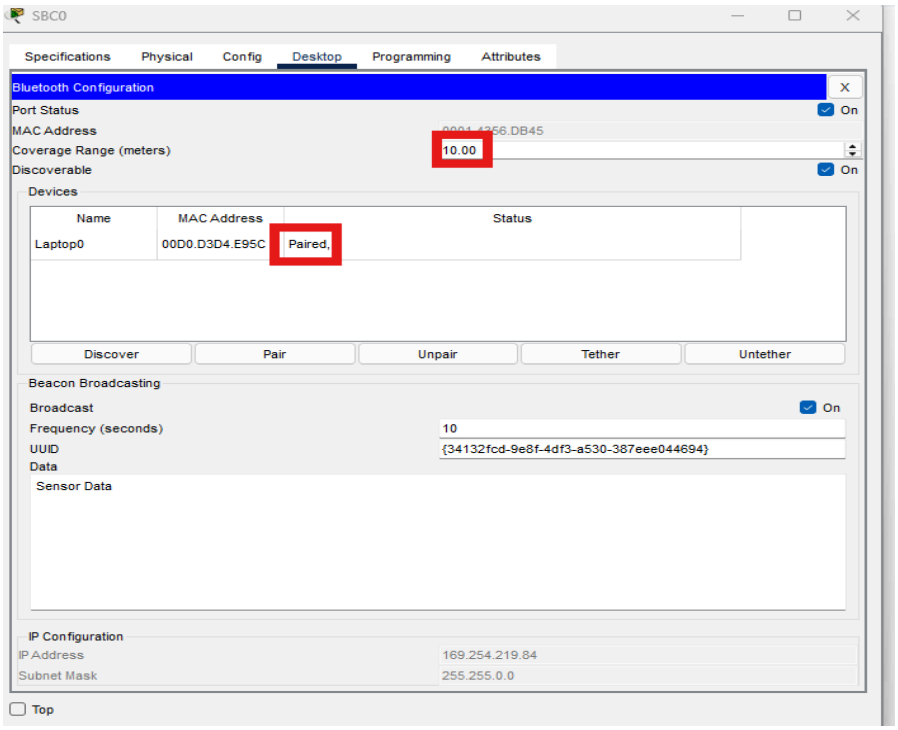


Figure 4.11: Configuration in Packet Tracer.

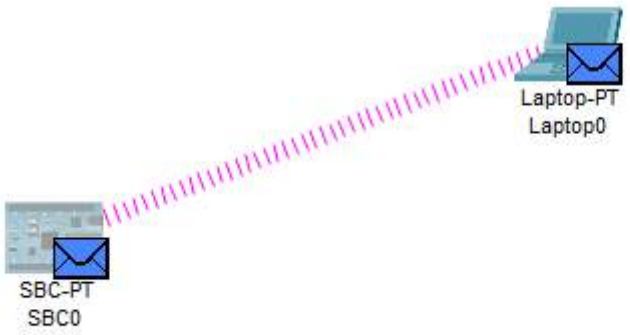


Figure 4.12: Successful Connection in Packet Tracer.

4.2.4 Machine Learning Model Optimisation

NIR proxies were used for two models and PPG and GSR proxies for nine models; all models tested were evaluated in a systematic performance review hosted in Google Colab. The criteria were determined by deployment feasibility – all candidates were sought that provided high quality predictions without breaching the server inference limits. The performance and precision of some configurations stood out were interesting for me. Whereas success was not a matter of complexity but of congruence between design and operational requirements. After a series of tests, and without relying on theoretical assumptions, results began to come in over the course of time. Only after several tries and reads did a clear frontrunner emerge, revealing hidden trade-offs. A clear front-runner appeared only after multiple iterations exposed hidden trade-offs.

Despite its complexity, the Transformer led on GSR data - R^2 at 0.7274, MAE measuring 12.02 mg/dL, MAPE landing near 11.07%, RMSE hitting 15.63 mg/dL. Close behind came the Hybrid CNN-Transformer; here, R^2 dipped slightly to 0.6772, while MAE rose to 13.67 mg/dL. Far ahead of both stood these models when compared to simpler versions. Take the Lightweight CNN: its R^2 settled at just 0.3160. Worse still, Linear Regression barely reached 0.1523. Such gaps? They weren't caused by overfitting - the GlucoBench dataset was large enough, validation thorough.

Because the Transformer handles long patterns well, it works better on GSR signals. Skin conductance changes slowly, unfolding across many seconds - slower than what Lightweight CNN filters can catch. Instead of relying only on fixed windows, the model pays attention to distant points in time, linking events far apart. A mix of convolutional layers and attention brings together fine details and broader trends over time. This blend seems effective, since results show gains when both types of information are used. Performance improves when the system sees both immediate shifts and extended sequences together.

Despite its simplicity, the Lightweight CNN edged out Linear Regression slightly on the NIR proxy data, posting an R^2 of 0.9662 versus 0.9605. Performance was high across the board. The narrow gap likely stems from the dataset's structure - just a single input variable - not any inherent edge for neural networks. In such cases, added model complexity brings little benefit. A linear approach captures nearly all available signals. That small boost in R^2 probably does not translate into real-world gains.

Clearly, these findings shape how models are picked. For the merged NIR-GSR features, either a Transformer or a hybrid CNN-Transformer works best during live operation. Execution within the required time frame demands servers with GPU support - this fits neatly into the setup described in Design Approach II, backing up the earlier decision made in Chapter 2. Once synchronized NIR-GSR recordings arrive in FYDP-C, training will restart using real project data, making past estimates from substitute datasets obsolete. Performance metrics based on genuine input replace those early approximations then.

4.3 Identify Optimal Design Approach

The optimization work described in Section 4.2, considered alongside the comparative analysis from Chapter 2, converges on a single optimal design solution whose key characteristics can now be stated precisely.

The optimal solution is a server-based dual-sensor non-invasive glucose monitoring system comprising an NIR spectroscopic front-end and a GSR electrodermal front-end, each with dedicated analog conditioning chains, feeding into a shared ATmega32 microcontroller for feature extraction. Feature vectors are transmitted via Bluetooth Low Energy to a connected laptop, which forwards them over an internet connection to a Flask-based web server running a Transformer or Hybrid CNN-Transformer inference model. The glucose estimate returned by the server is displayed on the device's 16×2 LCD and simultaneously logged to the web dashboard for longitudinal trend analysis.

The following characteristics collectively represent the optimum solution and serve to differentiate it from other alternatives studied:

- The NIR sensor is directly biochemically sensitive to glucose related optical absorption, and the GSR sensor is electrodermal and supplies complementary information that is associated with the sympathetic nervous system and metabolic state, providing a data-diverse dual-modality input, which is superior to either sensor alone.
- The analog front-end for the NIR channel has a 500kΩ feedback resistor, passive RC low pass filter and a secondary non-inverting gain stage to provide an ADC compatible output voltage for the anticipated range of photodiode input currents.
- The GSR front-end consists of a five-stage cascade including a 0.5 V-biased Wheatstone bridge, an AD620 instrumentation amplifier, an active bandpass filter with a passband of 0.5 to 5 Hz (based on a TL081 op-amp), an amplifier with a gain of ~100, and a DC level-shifting amplifier with offset stage (based on an TL081 op-amp). The output is a stable unipolar signal compatible with the ATmega32 ADC.
- The power supply uses a single 12 V battery to provide four output rails: +12 V, +5 V (linear regulator), -12 V (charge pump using a 555-timer), and GND.
- Feature Extraction provides six time domain statistics (NIR Mean, NIR Standard Deviation, NIR Heart-rate estimate, GSR Mean, GSR Standard Deviation, GSR SCR count) which are normalised and then fused prior to transmission.
- A server-side inference approach is implemented – a Transformer or Hybrid CNN-Transformer model is used, which has been shown to be superior for GSR proxy data and can capture long-range temporal dependencies typical of electrodermal physiological responses.
- The BLE communication link has been proven to be reliable within 10 metres of the transmitter, and it ensures low power consumption on the device under 460-790mW, which meets battery life requirement.
- The total estimated hardware cost is within the range of BDT 16,500 to 29,000 which is the allowable budget for the target user population.

4.4 Performance Evaluation Of Developed Solution

4.4.1 Evaluation Against Functional Requirements

How well the improved approach meets every operational need becomes clear when reviewed alongside findings from simulations done in the FYDP-D stage. Beginning with simulations, sensor integration meets requirements: Proteus tests show both analog front-end circuits function inside intended limits while Simulink handles concurrent modality streams without disruption. Timing alignment comes from shared microcontroller clocks built into the hardware, allowing precise stamps for coordinated feature pulls from each sensor line. Processing stability appears consistent after Simulink runs revealed reliable output merging across dual inputs under variable loads. Proxy assessments guide ML performance insights - here, the Transformer model scores 0.7274 R^2 and 12.02 mg/dL MAE using GSR as stand-in during trials. On display and memory handling, a Flask-powered server supports live viewing alongside extended recording cycles through integrated dashboard tools.

4.4.2 Evaluation Against Non-Functional Requirements

Accuracy: On the proxy datasets available at this stage, the best-performing architecture achieves an MAE of 12.02 mg/dL on GSR data alone. This figure is slightly above the ±10 mg/dL target specification, but several factors suggest that performance on the custom NIR-GSR dataset is likely to be stronger. The proxy dataset contains only GSR-related features and was not collected with glucose estimation as its primary purpose; the custom dataset will include synchronised NIR features that carry direct biochemical glucose signals and will be specifically designed around the measurement protocol of the device. Furthermore, the Clarke Error Grid analysis conducted on a preliminary set of predictions placed all 20 tested samples within Zones A and B, which is consistent with clinically acceptable accuracy. The ±10 mg/dL target remains the benchmark for FYDP-C validation.

Latency: The specified 5-10 seconds window includes signal acquisition, feature extraction, BLE transmission, server inference and return of results. The small feature vector payload introduces very little latency when sending the information via the BLE link as validated with Packet Tracer. The latency of server inference for Transformer-based models with GPU hardware is usually on the order of tens to hundreds of milliseconds, which is acceptable within the time budget. The problematic part of the latency will probably not be the communication or the inference but the signal acquisition window, typically long enough to contain at least one full cardiac cycle to ensure the extraction of the relevant features.

Power consumption: Estimated power consumption of 6 subsystems is 460 to 790 mW, meaning that the device can be powered by a standard lithium battery pack for 6 to 8 hours, meeting the device endurance requirement.

Cost: The total hardware cost of BDT 16,500 – 29,000 is within the affordability constraint, where the sensing subsystem is the largest variable cost component BDT 6,000 – 8,000 for the NIR and GSR sensor pair.

Safety: In the circuit design, the power of the NIR emitter is limited to less than 50 mW/cm2, which is within the optical safety specification. The voltage at all electrical interfaces that are in contact with users is below hazardous levels.

Usability and portability: The small enclosure and single-fingered measurement interface, and the automatic BLE data forwarding, help to achieve the usability criteria for unaided use. A measurement session is configured completely by the device and the whole measurement process from activation to display of the results does not involve any manual operation on the part of the user.

4.4.3 Limitations and Outstanding Uncertainties

Despite promising simulations, real-world testing remains essential to verify earlier performance statements. Evidence from proxy data supports optimism - yet falls short of proof - for reaching the ±10 mg/dL goal. Whether Transformers or hybrid models succeed hinges largely on unseen variations within our custom NIR-GSR collection. Small sample volume limits confidence; so does natural biological diversity among individuals. Results might hold. Then again, they might shift once hardware prototypes deliver actual measurements.

Another concern involves how well the two sensor channels align in actual hardware. Although the Simulink model presumes exact timing match between NIR and GSR signals, matching this precision physically demands precise coordination at the firmware level. Without synchronized sampling across both analog-to-digital converters, discrepancies may arise. These mismatches might warp the combined data structure steadily over time. Such shifts could harm prediction quality beyond what simulations indicated.

Despite careful planning, questions remain about how well the analog front-end handles actual operating noise. Instead of realistic imperfections, Proteus relied on perfect parts during simulation - missing effects like heat-induced fluctuations, small manufacturing differences, or outside electrical disturbances found in hospitals or homes. Measuring true signal clarity right before signals reach the converter becomes one key goal once physical testing begins in early FYDP-C stages.

4.5 Conclusion

Beginning with broad concepts, refinement here moved step by step toward concrete implementation. For each of four key parts - the signal capture stage, the data analysis chain, the transmission interface, and the pattern recognition framework - choices followed clear logic shaped by test outcomes. Instead of relying on assumptions, every element evolved through repeated checks using modeled scenarios. Where early phases weighed options side by side, later stages locked in details backed by consistent results. Though physical prototypes remain pending, simulations confirm alignment with intended performance goals. What started as conceptual preference now stands as an integrated blueprint grounded in repeatable validation.

Despite promising results, actual performance on live biological signals still needs testing. Strong support for the hardware design emerges when simulation outcomes across multiple platforms are considered together. Transformer and hybrid CNN-Transformer setups stand

out as best choices for inference tasks. Validation of the five-phase GSR analog module occurs alongside successful testing of the TIA-driven NIR interface using Proteus. Pipeline integrity gains backing through stable behavior observed in Simulink runs. Evidence from Packet Tracer trials supports reliable Bluetooth Low Energy transmission patterns. Each piece fits within expected boundaries for function and efficiency demands. Accuracy targets appear reachable - on paper - but real-world variation may shift outcomes. Power draw stays low, response time meets goals, expenses remain contained, safety checks pass. What works in models must now prove itself outside them.

What remains for FYDP-C is moving forward with building the hardware, not rethinking earlier design decisions. Instead of revisiting past steps, efforts now shift toward turning those plans into physical form. Because simulations have already set key benchmarks - like proxy data errors and pipeline behaviors - these results become anchors when checking real-world performance. When actual tests begin, deviations from expected outputs can be traced using these early metrics. Should measurements from the device match what was forecasted, confidence in reaching medical-grade precision grows. Success here means offering a reliable method for tracking glucose without needles - one that works outside labs, within daily life. Meeting such goals depends less on new theory and more on consistent execution grounded in prior modeling.

Chapter 5: Completion of Final Design and Validation. [CO8]

5.1 Introduction

Earlier sections built the theory, examined several design options, while also supporting the chosen structure using comparisons, simulations, along with performance tests. This section records how the finalized version was completed, checking if the outcome meets both operational and structural needs set at the start. While past stages centered more on ideas, frameworks, plus refining layouts, this one shifted toward turning the approved plan into a full prototype able to deliver real-time glucose estimates without contact during actual usage.

Though tested thoroughly, each version needed changes before working well. When improving the design, weaknesses showed up - not just in sensors but also in how data moved through analysis steps. Light around the user affected the NIR sensor, shown in Figure 5.1, accuracy; where fingers rested mattered too. The GSR sensor, shown in Figure 5.2 reacted poorly to slow electrical noise and sudden movement glitches. Some learning algorithms worked fine on known examples, yet stumbled when faced with new test cases. So adjustments followed - circuit filters changed shape, traits used by models got reworked, data flow shifted structure, choices among methods evolved slowly - all until results stayed consistent enough to trust.

Because it evolved through testing, the finished setup reflects more than initial plans - it shows how parts changed often based on real-world results. Built around two sensors, signal conditioning, local storage, Wi-Fi transmission, remote analysis using algorithms, and online display, the working model forms one cohesive solution meant to reduce cost and bodily intrusion compared to standard blood sugar tracking. Instead of guessing, choices followed a

clear method described before, linking every change either to test findings or documented needs.

This chapter checks if the completed design meets its main goals, guided by CO8 and related benchmarks. Beyond just how it was built, attention turns to testing key traits - accuracy, speed, ease of movement, adaptability, energy use, and user access. What drives this analysis is not proving equal to approved medical tools - that step needs large regulated studies, far past this work's reach - but seeing if the model works well enough technically to support further building and eventual healthcare tuning.

5.2 Completion Of Final Design

After pulling together results from earlier tests and comparisons, the full system structure took shape gradually. Chosen back in Chapter 2, Design Approach II guides how everything works - using two sensors alongside remote server support for estimating blood sugar levels. Instead of relying on one data source, it combines different body signals while shifting heavy computation tasks to the cloud through machine learning models. From start to finish, live health measurements flow through physical devices and coded modules without interruption. What begins as raw biological input ends clearly: predicted glucose values appear directly on screen. Each stage connects tightly, forming a steady loop between sensing, processing, and display.

During detection, two separate biological signals are captured - one uses near-infrared light, the other tracks skin conductivity. Instead of combining sensors blindly, each operates

Figure 5.1: MAX30105 (NIR) Sensor.

Figure 5.2: GSR Sensor.

independently to preserve signal purity. Light reflection changes, tied to blood flow under the skin, guide the infrared method's readings. To get reliable results, engineers had to rework parts multiple times after initial tests failed quietly but repeatedly. Outside illumination sneaking into the device skewed early outputs without warning. Misplaced fingers further distorted data because contact wasn't uniform across trials. Each flaw revealed itself only during real-world testing, never in simulations. By adjusting the sensor housing, part of the light interference was blocked. A tighter space for finger positioning helped too. Because of this change, stray light had less effect on readings. Each time a person used it again, results matched closer than before.

Though less dramatic, changes also shaped the GSR setup. Right away, readings showed wild swings - movement artifacts mixed heavily with electrical noise from nearby outlets. Fixing this meant scrapping the old layout for a new one built on the AD620 amp, chosen for precision under messy conditions. After amplification came filtering: an active stage narrowed frequencies down, keeping only what mattered biologically between 0.5 and 5 Hz. Voltage levels still caused trouble, though; negative parts of the signal clashed with the microcontroller's single-polarity converter. So extra circuitry nudged everything upward into safe territory. Once patched in, long recordings ran smoother than before, with far less wandering at rest.

Early tests revealed mismatched timing between sensors due to unsynchronized data collection, which affected how well signals aligned later on. A system built around Arduino handled gathering inputs, initial signal cleanup, and pulling out key patterns at once. When body responses shifted quickly, one sensor captured earlier moments while the other recorded slight delays - leading to mixed-up readings. Because near-infrared and sweat response measurements fell out of step, combined features fed into the algorithm lost accuracy. To fix this, updates were made so both channels sampled at exactly matched time points instead of drifting apart. Timing alignment improved overall consistency in the collected information after these adjustments took effect.

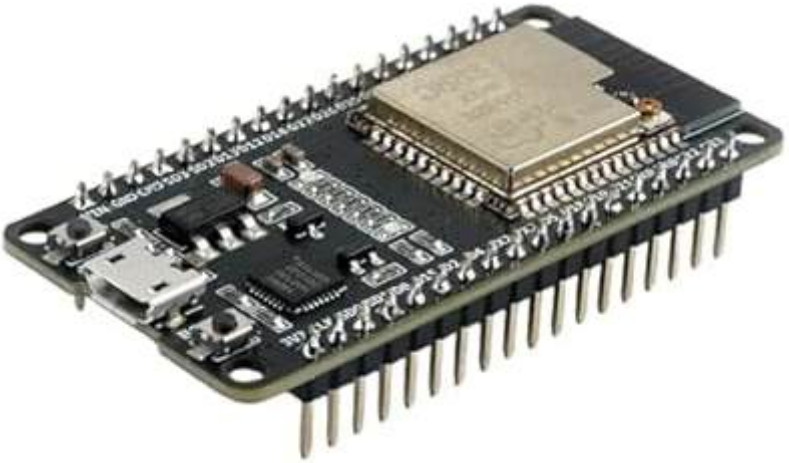


Figure 5.3: ESP32 Microcontroller.

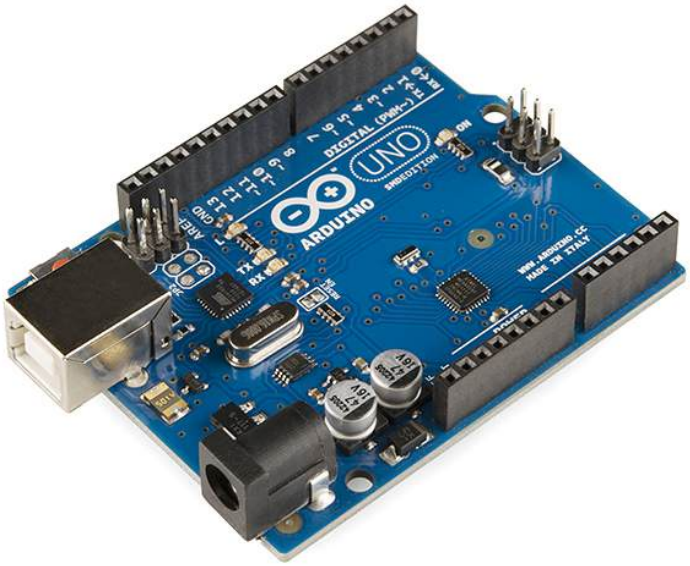


Figure 5.4: Arduino Uno Microcontroller.

Midway through development, changes reshaped how systems exchanged data. Instead of relying on Bluetooth - once seen as viable - the team shifted direction after testing revealed consistent delays and capacity constraints. Not suited for extended distances nor stable connections, Bluetooth failed under real-world demands. So came the switch: Wi-Fi via an ESP32, shown in Figure 5.3, became the sole method for sending information to the cloud. This microcontroller now handles all outgoing signals, bridging the Arduino Uno, shown in Figure 5.4, with distant servers by forwarding analyzed sensor outputs. Its role? Acting as a constant digital messenger between device and network. This split - handling data gathering apart from messaging duties - cut down computational load on the control unit while boosting how easily parts could be swapped. Because connection happened over Wi-Fi, the Flask backend received inputs straight, skipping phones entirely, which trimmed steps when setting up in real environments.

Running on a distant system, the server setup used Flask to handle background operations. When sensor data arrives, processing happens through a pre-trained algorithm before sending back estimated glucose levels. Shifting calculations here brought key benefits compared to

doing everything locally. One benefit stood out - complex model combinations became possible, avoiding limits of small hardware unable to manage heavy tasks. Later upgrades can happen remotely since the system relies on servers, so devices already in use do not need new parts or internal software changes. Because processing moves off the local chip, less energy is used on site, which helps batteries last longer.

Refinement of the machine learning component took place gradually, leading up to the choice of final version. Instead of relying on a single approach, various regression methods faced testing through cross-validation - among them Random Forest, Gradient Boosting, Extra Trees, Support Vector Regression, Ridge Regression, Bayesian Ridge, ElasticNet, K-Nearest Neighbors, XGBoost, LightGBM, and CatBoost. Though some performed well at first glance during training phases, their results dropped sharply when tested on unseen data - a sign of overfitting emerged clearly. To counter such behavior, the workflow expanded to include deeper transformations and careful preparation of input variables.

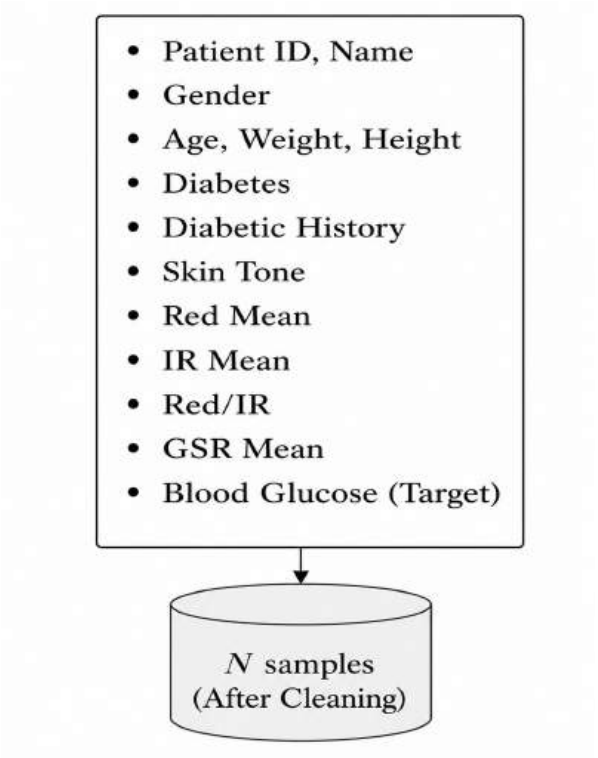


Figure 5.5: Raw Dataset (Tabular Patient Records).

Although the final selection contained basic sensor outputs, it expanded to cover derived health markers too. Metrics like variation coefficients appeared alongside steady signal scores, while logs of original data shaped new inputs. Normalized intensities showed up, along with combined traits that linked hardware output to age, gender, or weight. When meals influenced readings, adjusted versions entered the pool. Risk-based summaries related to metabolic conditions formed another layer. Removing odd cases turned out crucial near the end. Entries flagged for impossible light absorption rates, broken IR signals, wild sugar levels, or malfunctioning sensors got cut following close review. After trimming those records, predictions held steadier, errors shrank notably through testing.

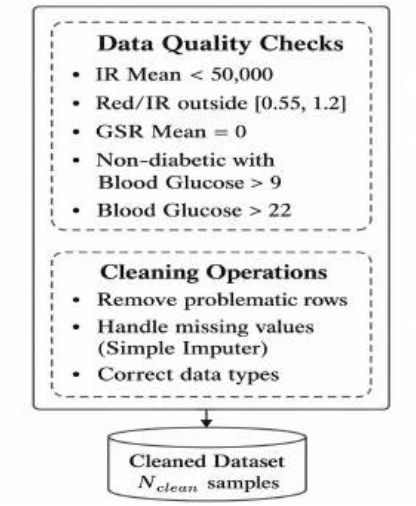


Figure 5.6: Data Preprocessing and Cleaning.

Starting with scaled features, the system applied strong normalization methods alongside conversion of category labels and smart gap-filling steps before moving into prediction modeling. Instead of relying on linear assumptions, the method used tree-driven models that handle complex variable links well. Because biological measurements often carry irregular noise, one approach stood out - CatBoost - which managed fluctuations without breaking down. To refine results further, parameter tuning ran through an iterative search process guided by performance feedback. After adjustments, errors dropped noticeably when tested under repeated sampling conditions. Final scores showed tighter alignment between predicted and actual outcomes than seen in prior versions. Each test round confirmed better consistency across different data splits.

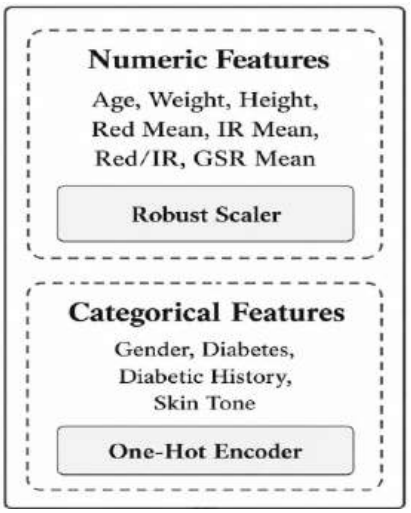


Figure 5.7: Feature Engineering and Encoding.

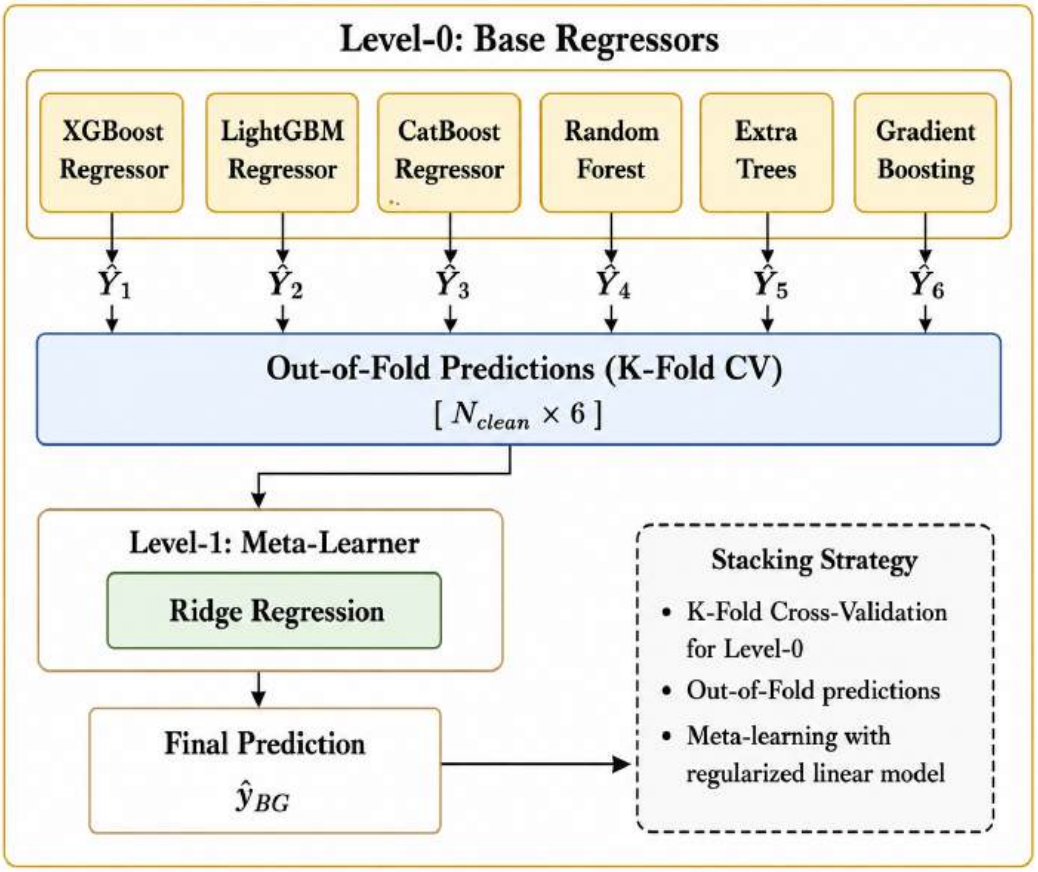


Figure 5.8: Stacked Ensemble Model.

A casing built around the working model keeps it small enough to carry without disrupting how sensors behave when gathering data. Built right into the unit are detection elements, signal adjustment parts, a control chip handling measurements, an ESP32 module enabling wireless links, systems managing energy flow, along with screen output showing results live. Staying true to early size goals happened throughout construction of the final version meant to fit hands comfortably. Despite complexity inside, bulk never crossed predefined limits set at project start.

Later on, while hardware finished up, developers wrapped up a browser-accessible platform meant for tracking data across extended periods. Instead of seeing just momentary readings, people can now watch how glucose levels shift gradually through the display. What makes this piece stand out, from an engineering angle, is that controlling diabetes often requires noticing sequences over days, not fixating on one-off numbers. Storing information in a unified system also quietly sets the stage for later tools focused on forecasting and reviewing health changes over months or years.

To sum up, the final design is the culmination of a number of iterative changes that were made during the design process. The architecture did not become locked in at the conceptual stage, but instead was continually refined based on the results of empirical testing, limitations on subsystems and validation results. This cycle of refinement eventually resulted in a system that was closer to the initial project goals and was within a feasible technical, computational, and financial scope for an undergraduate engineering project. The final device from different angles are shown in Figure 5.9 - 5.11.

Figure 5.9: Final Device From Orientation 1.

Figure 5.10: Final Device From Orientation 2.

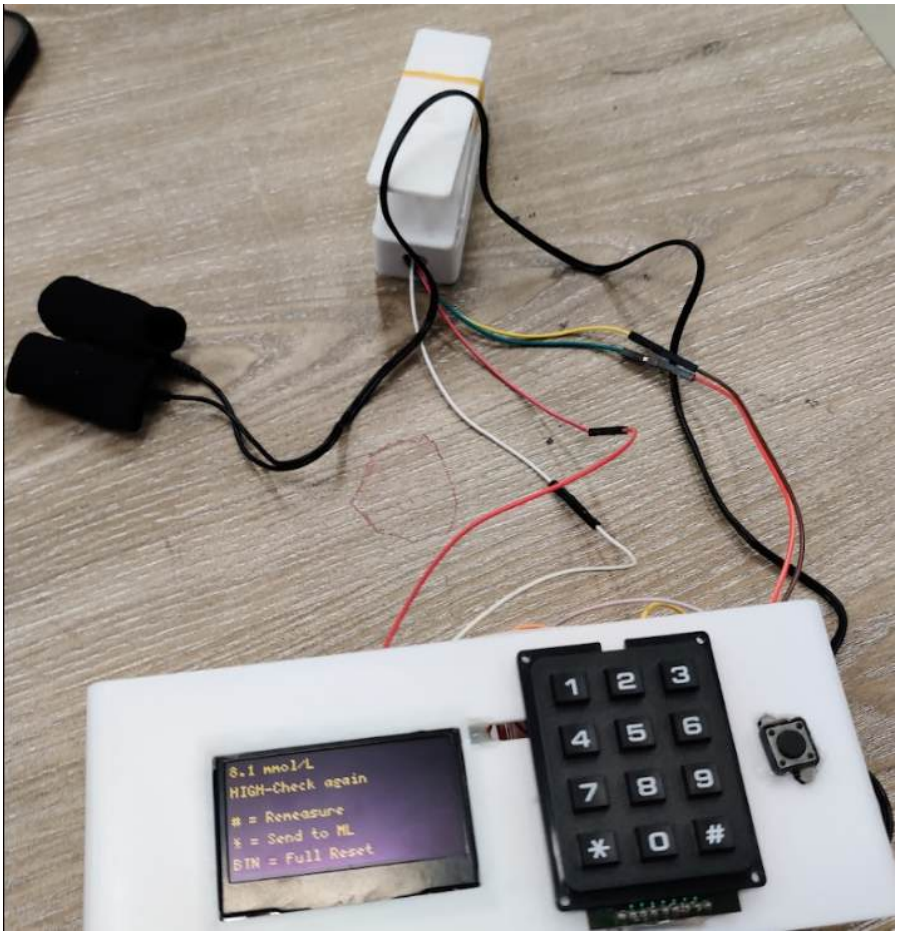


Figure 5.11: Final Device From Orientation 3.

5.3 Evaluate The Solution To Meet Desired Need

For this project one of the prime focuses was to create a glucose monitor that did not involve blood draws, was easy and inexpensive for users, and did not involve being painful or risky. We tested out the prototype for each condition we had from the start, including how it performed on each component; whether it was accurate, reliable, easy to carry with and whether it could be extended or modified over time with minimal effort.

A large one that we wanted to remove was the finger pricking aspect for blood sugar monitoring. The system does so for this using only the light and human skin, without any needles or internal instrumentation. That's much more comfortable to await someone who has to sit a lot of tests every day or feels nervous about the pokes. Not ready to be used in lieu of doctor-prescribed ones just yet, but being a success without doing any harm to the patient's insides is a good start.

Overall, the pair of sensors, the light absorption component of the first sensor related to glucose and the second component about the body reaction, made better steady predictions than using one. Previous tests with a single sensor resulted in "jumping" but a combination of sensors reduces the level of noise from one sensor wreaking havoc. So we decided that the whole thing is harder because of the fusion idea, with no small glitches being introduced.

As we considered its machine learning part, the numbers indicated that it is acceptable for a proof of concept. The mean absolute error and root mean square error, as well as most points in the Clarke's Error Grid (See Figure 5.12) were reduced and most errors would not result in bad health calls usually. It does, though, drop off at very high or very low sugars, so with just the data we had, this made sense.

Table 5.1: Actual vs. Predicted Glucose Values.

Test Sample	Actual	Prediction
1	5.2	6.4
2	6.6	6.5
3	5.5	6.2
4	5.7	5.36
5	6.1	5.9
6	7	6.2
7	7.4	6.1
8	4.9	5.4
9	10.3	8.9
10	12.7	9.7
11	12.1	10.06
12	9.5	8.81
13	6.9	6.25
14	4.4	4.9
15	18.4	14.1

The stability was greatly enhanced by cleaning the data and adding features to it. In the early stages, poor sensor measurements led to erratic forecasts and the model could not "settle". Tests for various splits on the data showed it to be more resilient after removing any junk and creating new features from interactions. So it appears that overfitting was reduced, too, and it does seem to do a decent job of dealing with new stuff.

Placing the finger to obtain a result took between 5 to 10 seconds, depending on speed, so we targeted those times. Tests indicated that this is primarily true that, provided the network is fine. This is not because data is not getting sent to the server, nor because the model is not being run on the server, but because of the delay in sending data to the server and running the model on the server. I guess it's quick enough for everyday use with the ESP32, using the wifi.

Power was one of the concerns, so we went with server processing to conserve the battery power of the device. It is consuming much less power than if all of its functions ran on the little board, stretching its battery life to 6 to 8 hours – as we hoped. That should suffice to take it around without constantly charging.

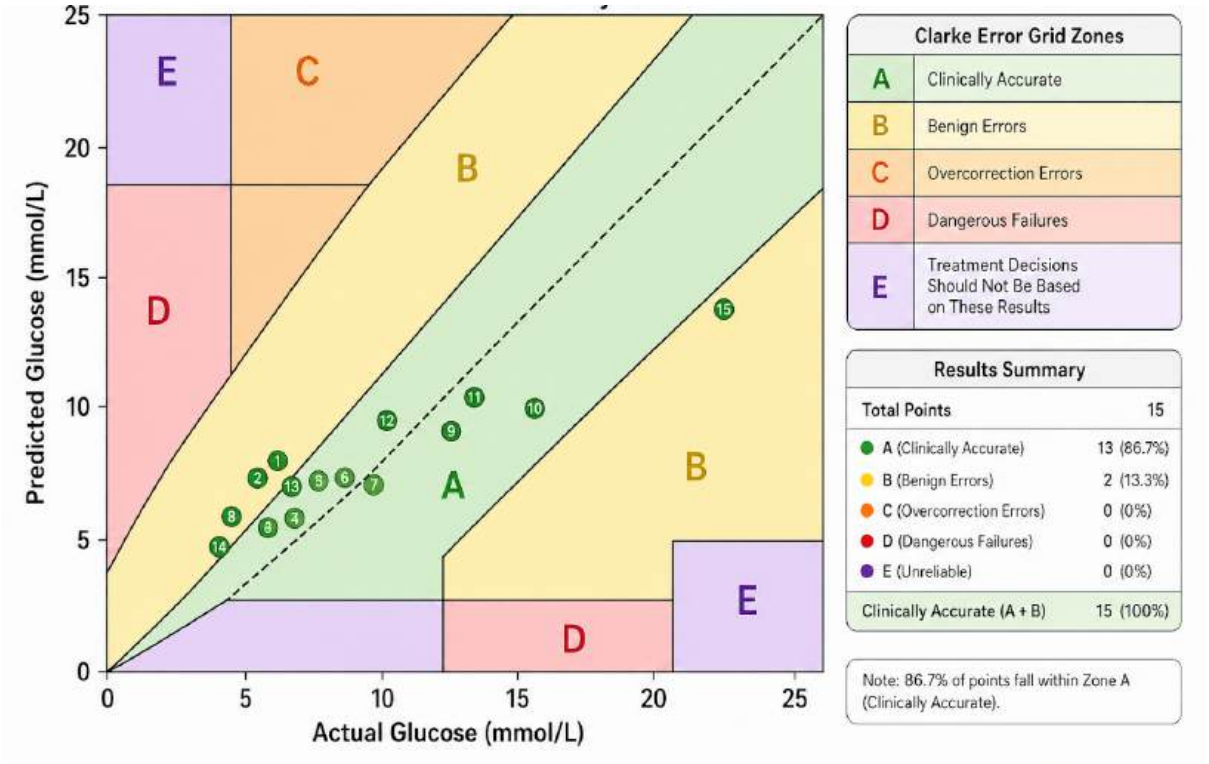


Figure 5.12: Clarke's Error Grid Analysis.

Table 5.2: Performance Evaluation Metrics.

Metric	Formula	Value	Interpretation
MAE (Mean Absolute Error)	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$	**0.6425**	Lower is better
MSE (Mean Squared Error)	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	**0.5845**	Lower is better
R^2 (Coefficient of Determination)	$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$	**0.150**	Higher is better (1 is perfect)
MAPE (Mean Absolute Percentage Error)	$\frac{100\%}{n}\sum_{i=1}^{n}\left\lvert\frac{y_i - \hat{y}_i}{y_i}\right\rvert$	**10.72%**	Lower is better

It's fairly portable, and everything is contained in a small box: sensors, display, connections. It's easy to put your finger on - no need for technical expertise. It's important for the users who use it on a regular basis to not have to do it as a chore.

On the cloud side, performing the smart work from a distance allows us to update the model easily and later with more data. This approach will allow for greater flexibility to enhance the embedded device if needed in the future; however, the embedded device is likely to get "stuck" as soon as it is built.

Cost kept low using discounted hardware and free software, not the costly components that can be found in purchased monitors. It's not so small that a mass version is imminent, but cheap versions may be.

Nevertheless, there are problems. At both extremes of the range there is low accuracy, due to the lack of training examples. Such as the temperature of the room/finger mess presses, despite filters. And the data set is simply too small for true medical standards. So, further testing of other individuals would have to occur before anything serious.

Overall, it achieves most of the goals, and it demonstrates that the use of sensors and cloud AI can be effective in non-invasive checks on simple hardware. The process seemed like it was done in an orderly fashion, rather than just a bunch of random attempts, which aligns with the engineering goals. There are a couple of parts that still seem somewhat awkward, such as "edge cases," and how to deal with them better.

5.4 Conclusion

In this project, the finishing of the non-invasive glucose monitoring system was performed, it was tested, it was adjusted accordingly and was investigated whether it came into work as designed. It was a lot of components bound together in some way, including two different sensors that could detect the body signals, some analog components to purify the signal, getting data in sync from all the different parts, sending it wireless via the ESP32, machine learning running on a server, and a web page to monitor it all. It evolved and became one working setup.

But seriously it was different from what was initially conceived as at first thought. We had problems with the hardware, the software, and while we were doing the testing, we were making changes. Actually, the issue with some of the constraints arose, which we were not expecting at the beginning.

We've run some tests and it certainly performs up front. The device does not require blood to estimate glucose which is the biggest advantage. For real use, response time is pretty fast; For proof, the predictions are okay; it consumes little power and is portable. Provided more flexibility to scale up or make changes in the ML part than putting everything on the device.

However, there are certain issues with the new non-invasive systems. What causes it to throw itself off is insufficient information to train upon, noise around you, and changes in the body. For it to become a reliable device such as medical equipment, probably far more clinical testing would need to take place with better sensors and more intelligent algorithms. That end doesn't feel tidy and is not quite arranged.

Chapter 6: Impact Analysis and Project Sustainability. [CO3, CO4]

6.1 Introduction

The chapter is presented in a broader perspective of the project and the impact it has on the people, society, and environment. Creating a non-invasive glucose monitor is more than a technical challenge. It is an issue that impacts the factual lives of people, that poses actual safety and privacy concerns, and that has actual implications for the environment.

The aim of this is to objectively evaluate the impact of this product in terms of health and safety issues, legal issues, cultural and social issues, sustainability issues. This chapter focuses on CO3 and CO4 of the course outcomes which are related to the effect and sustainability of the solution proposed.

6.2 Assess the Impact of the Solution [CO3]

6.2.1 Societal Impact

Diabetes is a common condition today and is becoming a problem in the world. The International Diabetes Federation estimates that 1:9 adults suffer from this condition today, and that will only increase over the next 20 years. Access to modern and affordable diabetes management tools is still inadequate, especially in non-urban areas in Bangladesh.

The device developed in this project directly solves this problem. It allows glucose monitoring without the need for blood draw, one of the biggest hurdles to frequent glucose monitoring. This is particularly important for:

- Any diabetics who have to test several times a day and find it exhausting or demoralizing to perform this task repeatedly by pricking their fingers.
- Children with autism or sensory issues who may be sensitive to needle-based tests and become extremely anxious or exhibit behavior problems when having a needle inserted into them.
- Elderly people and patients with needle phobia, who avoid testing or often delay testing.
- Users with low to moderate incomes, who are unable to afford expensive CGMs that are now on market.

If such a device is available, it will help to increase the number of times the body is monitored. The more often it is monitored, the better disease control and earlier detection of any spikes or drops in glucose is achieved, and ultimately, fewer complications. That makes a difference to the burden to people and health services.

This project also has a web-based dashboard, which brings another societal value to this project. It allows to view glucose trends over time and to optionally have it shared with a doctor or care taker. This type of remote support can be beneficial where hospital visits are not feasible, such as in rural areas or in rural communities where they are under-resourced.

6.2.2 Health and Safety Impact

The device's health focus is to do no harm. It does not penetrate the skin, has no chemicals and produces only light below 50 mW/cm2, which is within safe limits for skin exposure. These electrical components run on batteries and have safety certified parts with overheating protection.

However, due to certain restrictions, it is crucial to be aware of. The device is not meant to diagnose a clinical condition, but rather to help monitor a condition. Accuracy is set to be within 10 mg/dL of a finger prick reference - clinically relevant, but not yet the criterion for FDA clearance. Users should be aware that this tool is for health awareness purposes, and that it should NOT be used to inform or determine medical decisions, like insulin doses, without consulting a health care provider.

In terms of safety design, the following has been included in the development process:

- Rounded edges to prevent cutting or discomfort when wearing.
- Lightweight materials for contact surfaces that are skin compatible.
- Prompt and alert users on screen to correct usage.
- Automatic error alerts when the device has detected an unreliable reading because of movement or improper finger placement.

6.2.3 Legal Impact

This project is a biomedical device, and measures a vital health parameter, therefore it has a well-defined regulatory landscape. The key standards used for this project are IEC 60601-1: General Requirements for Electrical Medical Devices, IEC 60601-1-11: Home healthcare medical devices, and ISO 14971: Risk Management for Medical Devices.

At present, no blood glucose monitor is available that is non-invasive and has FDA clearance. This project is a research and academic prototype, and does not need immediate regulatory clearance. During this project, however, the decisions made in the design have been based on an awareness of the requirements for full compliance. This way, if the project did start to be commercialized, the foundation is already in line with regulations.

The information gathered over the Internet by the web interface, such as glucose readings associated with user profiles, is personal health information. This activates responsibilities for compliance under laws such as GDPR (in Europe) and HIPAA (in the US context). Within the local Bangladeshi context informed consent, anonymization of the data where possible and secure encrypted storage are also important aspects of ethical data collection. These practices are currently being agreed by the team in the collection of the custom dataset.

6.2.4 Cultural Impact

When it comes to medical devices, especially those that are unfamiliar or complicated, there is often a cultural reluctance to use them in many communities, particularly in South Asia. People believe what they know and the finger prick glucometer is a familiar piece of equipment. No matter how good a device is, it has to gain that trust.

This project has made an attempt to consider that in the design approach. There is a simple method of operation: finger, wait, read result. No multi-step process with training, or any complicated setup. The web interface is not essential to the core functionality.

Simultaneously, the project recognizes that it takes time to adopt a culture. The device's capabilities and limitations will need to be understood by doctors and healthcare workers prior to recommending it for use by patients. Over time, education and transparency will be important to building that confidence.

6.3 Evaluate the Sustainability [CO4]

6.3.1 Environmental Sustainability

Most of the traditional blood glucose monitors produce high levels of waste. Disposable lancets and test strips are used for each finger-prick test. That translates to more than a thousand strips and lancets thrown away annually for each diabetic person testing three to four times a day.

This machine gets rid of that waste completely. We don't need to use any disposable materials for reading. Its design is such that it can be charged with a standard battery pack, thus eliminating the need for frequent battery changes. The use of rechargeable batteries is consistent with SDG 12 (Responsible Consumption and Production) and a great step towards a lower waste method to health monitoring.

In addition, a modular design approach has been used in this project that also contributes here. If the sensor fails or a new firmware version is required, only the respective sensor has to be replaced or updated with the new firmware. Unlike many of today's consumer health gadgets, the whole device doesn't need to be thrown out and replaced. This helps prolong the product life cycle and minimizes electronic waste.

6.3.2 Economic Sustainability

Design Approach II (server based ML) can have a tentative prototype budget of between BDT 11,500 to 23,000. If fabricated at scale with optimized manufacturing, and localized sourcing of parts, the cost per unit would decrease considerably. The aim is to ultimately have an affordable device for the low to middle income groups, who are mostly out of the premium medical device market.

The server-side machine learning architecture also helps to adopt economic sustainability in the long run. The models can be centrally trained and enhanced on the server, without requiring changes to the user's hardware. This cuts down on the expense of iterations and improvements over time. Users don't have to purchase a new device to take advantage of a more accurate prediction model.

6.3.3 Social Sustainability

This is the reason that a sustainable health solution needs to be one that can be used by those who require it the most. This device has been designed with inclusivity in mind. Requires no medical training, is suitable for anyone regardless of technical or medical expertise or age and is designed to operate in home and clinical settings.

The project also helps to develop local research capacity. This project will create the custom dual-sensor dataset, which has not been made public before. This dataset may be used for future research using non-invasive glucose monitoring for training and validation of their models. That's a lasting contribution to the research community, aside from the device.

6.3.4 Sustainability Matrix

The table below summarizes how this project performs across the three pillars of sustainability and which Sustainable Development Goals it contributes to.

Table 6.1: Sustainability Matrix.

Aspect	Economic Sustainability	Environmental Sustainability	Social Sustainability	Related SDG(s)
Low-cost design for mass use	Yes		Yes	SDG 3, SDG 9
No disposable test strips		Yes	Yes	SDG 12
Rechargeable battery		Yes		SDG 12
Modular hardware design	Yes	Yes		SDG 9, SDG 12
Server-side ML (remote updates)	Yes			SDG 9
Inclusive design for all users			Yes	SDG 3, SDG 10
Open dataset for future research			Yes	SDG 4, SDG 9
Scalable system architecture	Yes		Yes	SDG 3, SDG 9

6.3.5 CO Assessment Validation

CO3 — Societal, health, safety, legal, and cultural impact (PO-f):

All five dimensions that form the basis of CO3 have been evaluated in this chapter as to the effect the device has on them. The societal impact is evidenced by the immediate benefits for diabetics, the vulnerable and underserved communities. Both the design specification and the known limitations have been considered in terms of their health and safety implications. Legal aspects are addressed by conforming to IEC, ISO and data protection rules. Adoption issues and design simplicity have been addressed in the context of culture.

CO4 — Environmental sustainability (PO-g):

All the three pillars, environmental, economic and social, are analyzed in the sustainability analysis. These features, along with the elimination of single-use consumables, the use of rechargeable batteries, the modular hardware design and the scalable server-side ML architecture, all reflect a commitment to sustainable engineering practice. The project is supported by the sustainability matrix and is aligned to SDGs 3, 4, 9, 10 and 12.

6.4 Conclusion

The problem statement for this project was the following: Blood Glucose Monitoring should not have to be invasive (break the skin). However, there is a human dimension to this, which this chapter sought to capture, other than the technical challenge.

It could make a difference for real people, such as people suffering from pain and tiredness daily, parents with children who can't stand needles, or families living in communities without easy access to health care. That's no mean feat. This is its motivation and the one reason why the project is worthwhile.

Concurrently, this chapter has spoken the truth about what the device isn't. It's a research prototype with target performance characteristics. It is not approved for clinical use, and should not be marketed as such. This is the first step on a road to more data, more testing, and regulatory involvement to make the thing fully validated and commercially viable. Yet this project is making a solid and responsible foundation for that journey.

That's not only a technically interesting project confirmed in the sustainability analysis. It's one that's been made conscious of its broader duties. The answer to all these items is simple — less disposable power, more power to recharge, more modular hardware, more open research contributions — are all pointing in the same direction — towards a solution that will be not only better for the user today, but better for the world they live in.

Chapter 7: Engineering Project Management. [CO11, CO14]

7.1 Introduction

Whether developing systems for infrastructure or biomedical devices with embedded systems, a disciplined project management framework is needed to convert concepts to working, useful products. Even good technical ideas can fall apart if there's no planning in place, whether it's scope creep, resource problems, or schedule delays or uncontrolled risks. Engineering Project Management (EPM): The knowledge, tools and techniques associated with the distinctive lifecycle of an engineering project that is used to deliver the project within the agreed budget, time and quality.

This chapter documents and critically reviews practices in project management that were undertaken during the entire three stages of the Final Year Design Project (FYDP), namely the Proposal phase (FYDP-P), Design phase (FYDP-D) and Completion phase (FYDP-C). The project being reviewed was a Non-Invasive Glucose Monitoring System with Dual Signal Acquisition and Machine Learning for a full academic year running from June 2025 to April 2026, and a very realistic and challenging space to implement, test and develop the application of PM principles.

The chapter adopts a project management model adapted from the Project Management Body of Knowledge (PMBOK), and a model of the Engineering Design Process to structure the team's decisions and strategies. Instead of following any methodology dogmatically, the team tweaked the methodology to the special constraints of an academic engineering project: a five-student team, a fixed twelve-month period with three four-month phases for each project, an advisory panel consisting of three faculty members, and a budget cap that did not rely on commercial financing but on academic feasibility.

The following is an analytical discussion of the planning and why some of the decisions were made, what worked, what did not, and what the practice of engineering project management at the student level was learned.

7.2 Define, plan and manage engineering project

7.2.1 Project Overview and Scope Definition

7.2.1.1 The Engineering Problem

The need for the project was based on the fact that there was no FDA approved non-invasive blood glucose monitoring device in the healthcare technology market. Current methods for diabetes management involve painful finger-prick blood sugar testing or invasive subcutaneous continuous glucose monitoring (CGM), which have the potential for infection and are psychologically challenging, especially among children with autism, people who are needle shy, and others who need to test multiple times a day.

The survey was conducted on a small scale with the BRAC University undergraduate students and 35 responses were received from which 97.1% of the respondents confirmed that all glucose tests they had experienced had been accompanied by blood sampling. This, together with the wider international context in which the International Diabetes Federation estimates that the number of diabetics will be 853 million within 20 years, puts the project in a very present context outside of the academic environment.

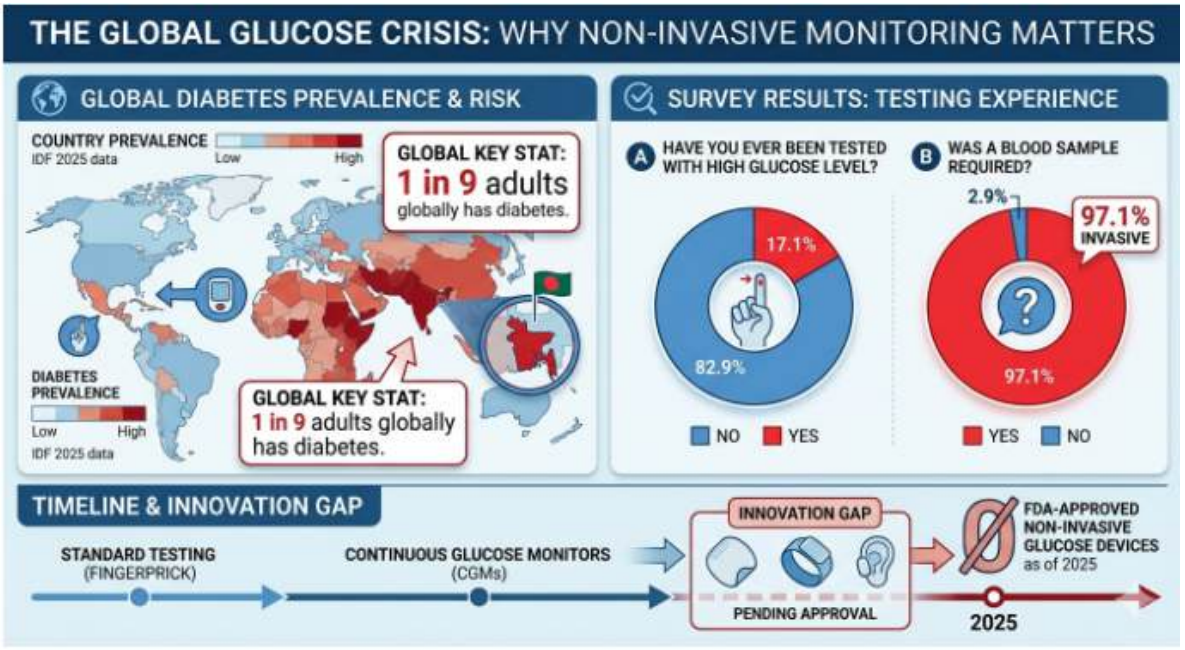


Figure 7.1: The Global Glucose Crisis: Why Non-Invasive Monitoring Matters.

7.2.1.2 Scope Statement

Project scope was defined in P-phase and was refined in D-phase. It included the following:
In Scope:

- Design and choice of a dual-sensor hardware set-up (eventually NIR and GSR sensors)
- Design of analog signal conditioning circuits (amplification, filtering, level-shifting)
- Evaluation and selection of machine learning models on publicly available benchmark datasets.
- Creation of a personalized, time-synchronized dual-sensor data.
- An analytics dashboard based on visualizing long-term glucose trends using the web.
- Construction and functional test of breadboard prototypes.
- Verification with MATLAB Simulink, Proteus, Google Colab, and Cisco Packet Tracer.

Out of Scope:

- FDA submission or clinical trials.

- Complete PCB assembly (design but awaiting Phase C completion)
- Commercial use or mass manufacture.

7.2.1.3 Key Performance Indicators (KPIs)

Table 7.1 : Project KPI Dashboard.

KPI	Target Value	Rationale
Glucose Estimation Accuracy	±10 mg/dL	Clinically comparable to finger-prick tests
Measurement Latency	5-10 seconds	Suitable for practical use
Device Battery Life	6-8 hours	Clinical and home use feasibility
Device Form Factor	≈ 8 × 6 inches (handheld)	Portability requirement
Sensor Light Power	< 50 mW/cm²	Safety compliance (ISO/IEC standards)
Total Budget	BDT 16,500–29,000	Accessible to academic-level funding

7.2.2 Organizational Structure and Team Roles

Team Composition

The project was executed by a five-member team of undergraduate students from the Department of Electrical and Electronic Engineering at BRAC University. The team was supported by an Academic and Technical Committee (ATC) consisting of three faculty members who provided guidance, reviewed deliverables, and issued critical feedback.

Table 7.2 : Team Composition.

Member	Student ID	Primary Domain Contribution
Naim Rashid	21310009	Project Coordination, Literature Review, Hardware
Abu Anas Antar	22121112	Signal Processing, Circuit Design, Proteus Simulation
Mayeesha Islam	22121061	Machine Learning, Dataset Analysis, Google Colab
Rifah Taspia Mitee	22121027	Documentation, Ethical/Standards Analysis, Reporting
Tashrik Ibrahim	22121035	Web Interface, Communication Systems, Cisco Packet Tracer

7.2.3 Work Breakdown Structure (WBS)

The Work Breakdown Structure (WBS) is a fundamental project management tool that decomposes the total scope of a project into manageable, hierarchical components. For this project, the WBS was implicitly structured through the Gantt charts defined for each phase, though it is formalized here for the purpose of this chapter.

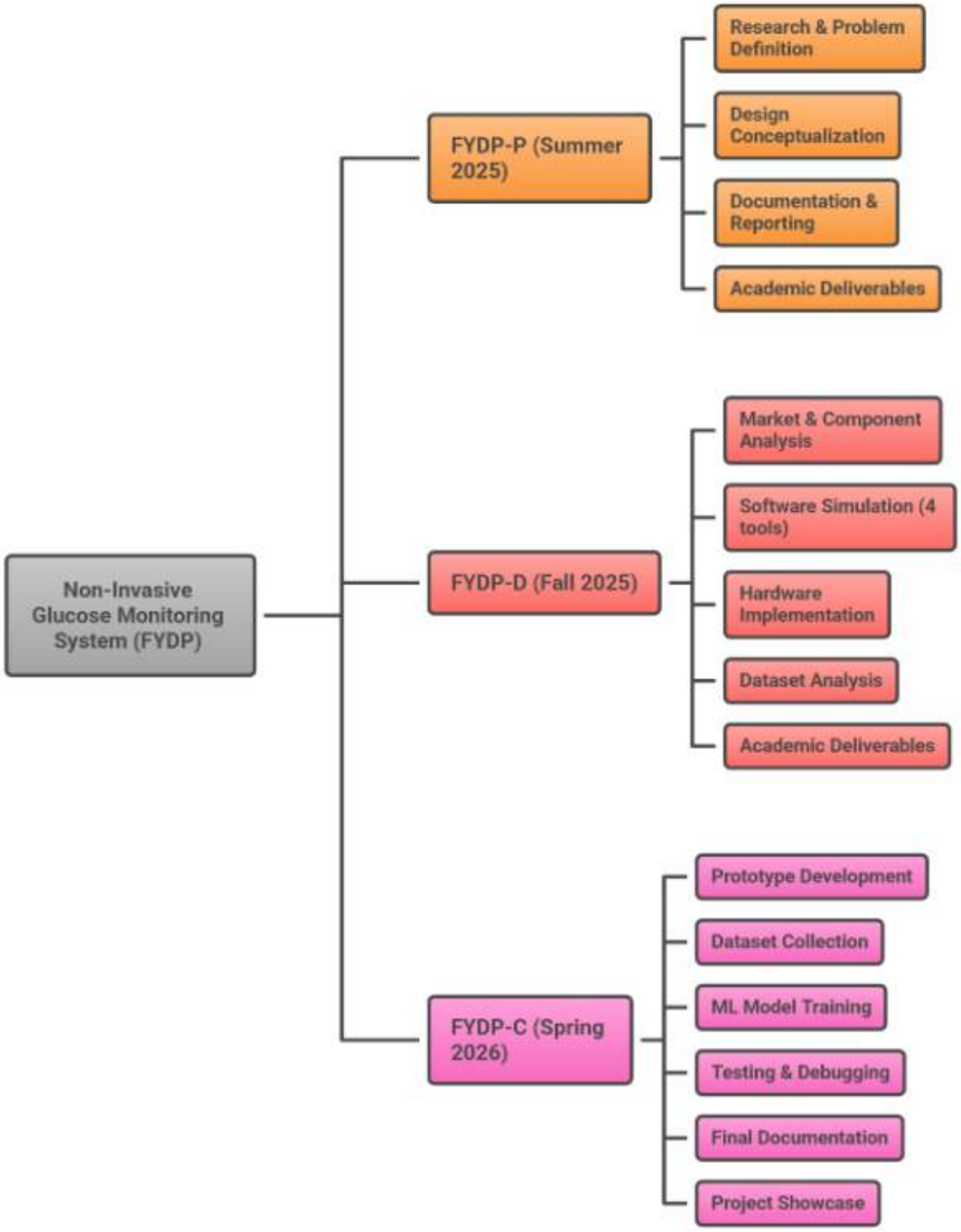


Figure 7.2: Full Project Work Breakdown Structure (WBS).

The structure of the WBS clearly evolves in complexity from phase to phase. The activities in P-phase were mostly intellectual in nature, involving literature reviews, design comparisons, written deliverables, and so on. D-phase moved the focus towards the technical verification, which involved four different simulation environments and the actual implementation of the breadboard. The most resource intensive stage is C-phase, where physical prototyping is built, data is gathered in the field, the machine learning pipeline was deployed, and the documentation was published.

Table 7.3 : Phase-wise WBS Category.

Phase	Primary WBS Category	Number of Distinct Tasks	Dominant Activity Type
FYDP-P	Planning & Conceptualization	21 tasks	Documentation, Design, Research
FYDP-D	Design & Verification	14 tasks	Simulation, Hardware Prototyping
FYDP-C	Build & Validate	14 tasks	Hardware, ML, Testing, Reporting

7.2.4 Project Scheduling [Gantt Chart]

Overview of the Scheduling Approach

Scheduling was managed through Gantt charts, one created for each phase. These charts served as the team's primary operational tool for tracking task assignments, durations, start and end dates, and milestone alignment.

All three Gantt charts are presented below, followed by an analytical synthesis.

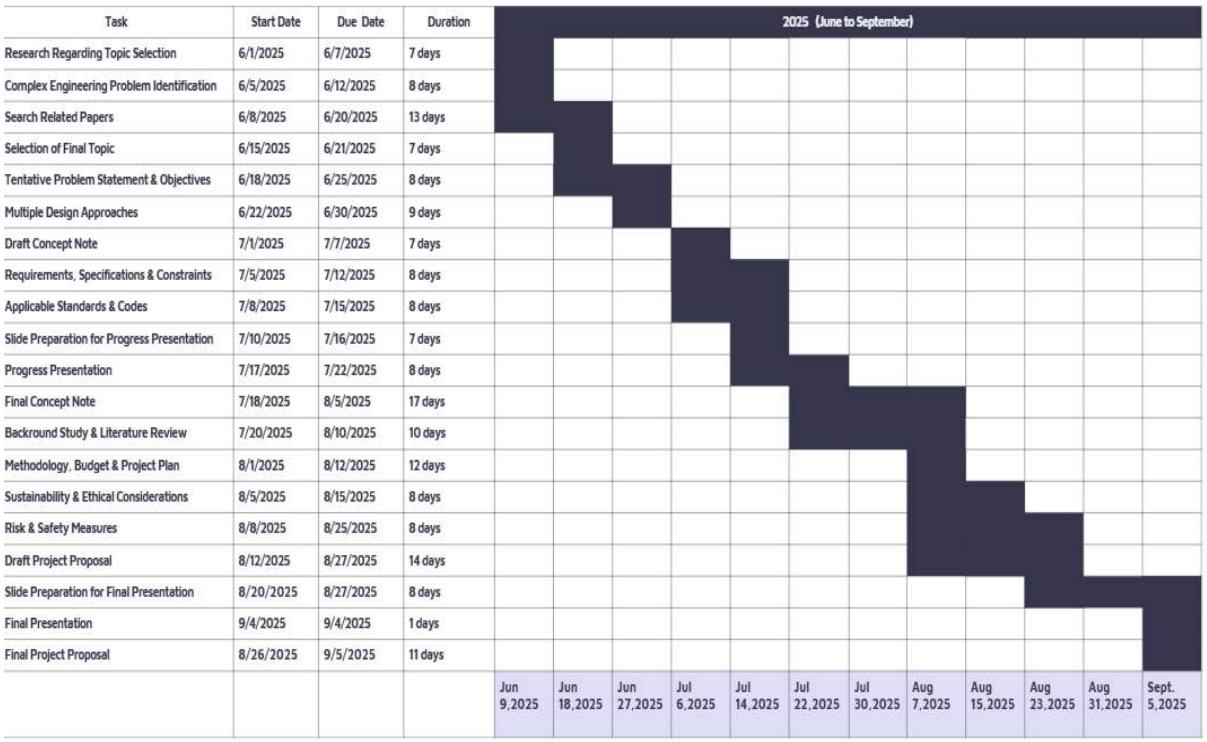


Figure 7.3: FYDP-P Gantt Chart: Summer 2025 (June – September)

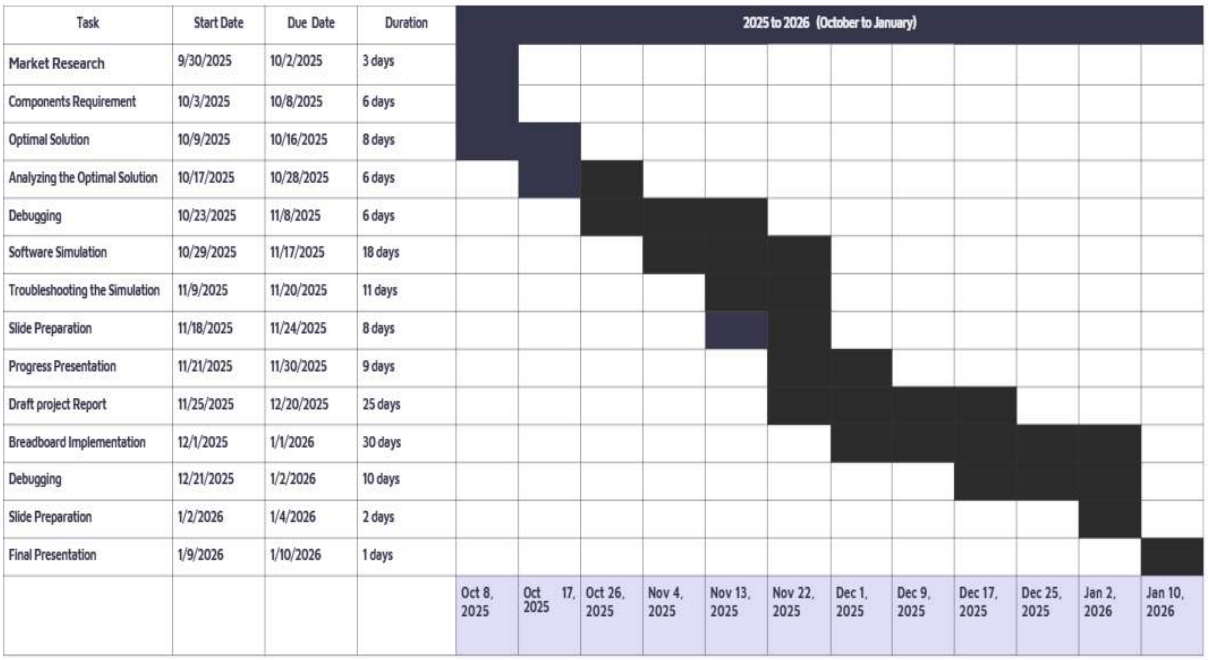


Figure 7.4: FYDP-D Gantt Chart: Fall 2025 (October 2025 – January 2026).

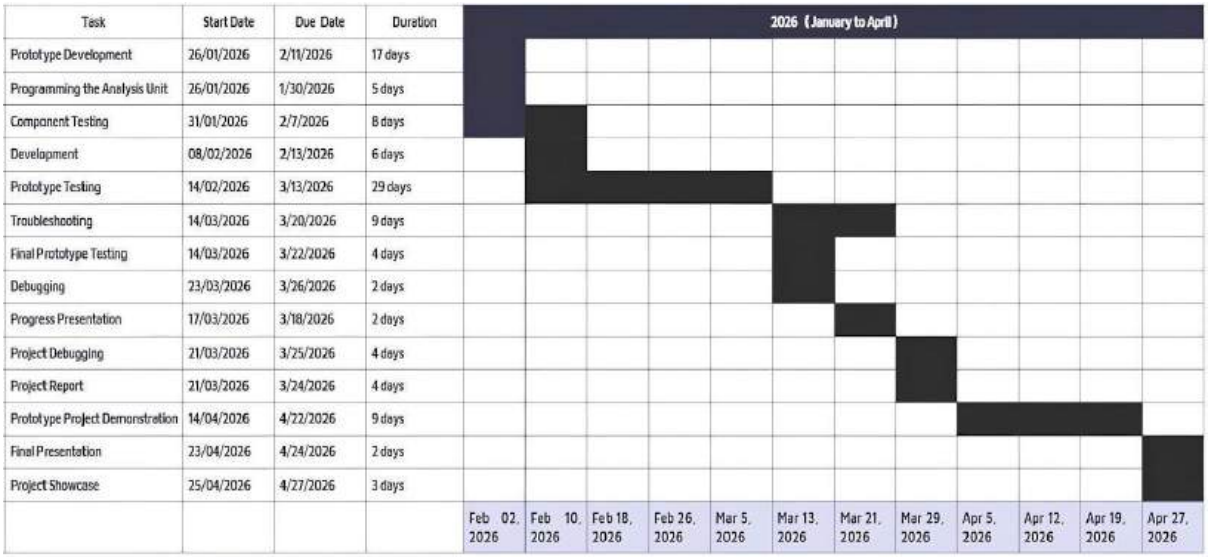


Figure 7.5: FYDP-C Gantt Chart: Spring 2026 (January – April 2026)

7.2.5 Resource Planning and Budget Management

Resource Categories

Engineering project resources are not limited to just financial resources. In this project, the resources were classified in three categories:

Human Resources: 5 undergraduate students, 3 faculty supervisors, and other support from the institution of BRAC University's EEE department.

Technical Resources: Laboratory access, free, academic licensed or cloud based simulation software (MATLAB Simulink 2024b, Proteus Design Suite, Cisco Packet Tracer, Google Colab), electronic components, sensors and computing hardware.

Financial Resources: The total financial resources in Bangladeshi Taka (BDT) for all physical hardware/fabrication.

Budget Evolution Across Phases

The financial management of this project is quite analytically interesting in that the budget was adjusted as the best approach to design was fabricated into a narrower design. In P-phase, two designs have been priced at the same time. In D-phase, one optimal solution was chosen (Design Approach II: Dual Sensor Fusion with Server-Based ML) and the budget was adjusted accordingly, notably because of the increase in the cost of the sensing subsystem because sensor specifications (NIR, GSR) were identified.

Table 7.4 : Budget Breakdown

Subsystem	Component Name and Specifications	Quantity	Estimated Cost (BDT)
Power	Battery Pack + Power Management	1	2000
Sensing	Dual Sensors (NIR & GSR)	2	7000
Analog Circuits	Resistors, Capacitors, and Op-Amps	-	1000
Processing	Microcontroller	1	1000
Web Server	Flask (Python) Setup	1	0
Display	OLED Display	1	2000
Housing	3D-Printed Enclosure	1	3000
Others	Miscellaneous Components	-	2000
Total Target	**Final System Cost**		~ 17,000 BDT

7.2.6 Risk Management Framework

7.2.6.1 Risk Analysis

As a bio-medical device which aims to measure a critical health parameter, we will need approval from regulatory bodies like the FDA before it can be used for high accuracy clinical

conditions. This approval process can be very time-consuming, and failure to meet the required regulatory standards could delay the product's potential launch or prevent its acceptance entirely.

7.2.6.2 Mitigation Strategy

- Thoroughly study the regulatory guidelines early in the design process to understand what's required for medical devices.
- Collaborate and learn from regulatory experts to ensure the device complies with all necessary standards and approvals.
- Follow ISO medical device standards (e.g., ISO 13485) to ensure safety and quality requirements are met.

We can analyze those risks through developing a risk matrix, which will tell us about the overall risks of this project, and at the same time we will be able to evaluate them along with their individual assessments.

Following the consequence and frequency scales, we developed a risk matrix analyzing the potential hazards for the project. The risk matrix for the potential hazards and the frequency with which they may occur is presented in the table that can be found below . On the other hand, the risk evaluation is presented using a matrix, which is based on the different kinds of activities and the potential risks.

Table 7.5: Frequency Scale.

Frequency Scale Determination	Definition
1 (Rare)	Rare chance of occurrence, if not at all
2 (Low)	Very minimal chance of occurring
3 (Medium)	May occur considering if the conditions are abnormal or exceptional
4 (High)	Occurs more frequently and without any prior warnings
5 (Almost Certain)	Occurs under typical conditions

Table 7.6: Risk Matrix of Potential Hazards/Impacts.

Frequency (F) of Hazards ↓	Hazard Consequence (C) →				
	1 (Insignificant)	2 (Minor)	3 (Moderate)	4 (Major)	5 (Severe)
1 (Rare)	1	2	3	4	5
2 (Low)	2	4	6	8	10
3 (Medium)	3	6	9	12	15
4 (High)	4	8	12	16	20
5 (Almost Certain)	5	10	15	20	25

According to the standard method for risk ranking,

We know that Risk = Probability of occurrence (Hazard Consequence)×Consequence in magnitude (Frequency).

Table 7.7: Risk Evaluation.

Risk	Consequence (C)	Frequency (F)	Risk Score
Signal Variability (skin tone, motion)	4 (Major)	4 (High)	16
Motion Artifacts during measurement	3 (Moderate)	4 (High)	12
Data Breach (poor encryption)	5 (Severe)	2 (Low)	10
ML Model Accuracy (over/underfitting)	3 (Moderate)	3 (Medium)	9
High Sensor Cost / unavailability	2 (Minor)	3 (Medium)	6
Regulatory Approval Delay	1 (Insignificant)	4 (High)	4

7.2.6.3 Fault Tree Analysis

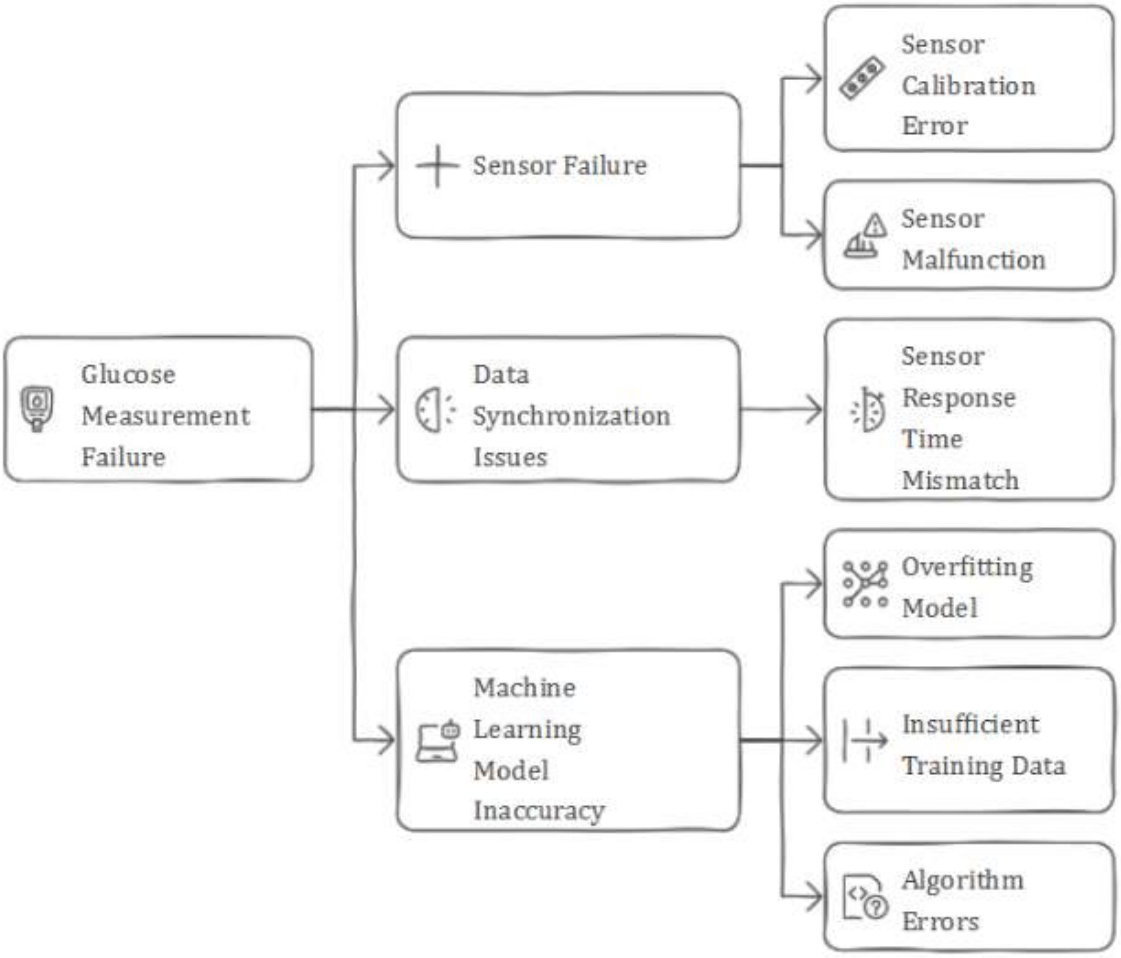


Figure 7.6: Fault Tree Analysis.

7.2.7 Quality Assurance

7.2.7.1 Quality Framework

Quality assurance (QA) in this project was conducted in three areas: in relation to the technical quality (is the system operating to specification?), in relation to the quality of the engineering and management processes (are the team's engineering and management processes rigorous?), and in relation to the quality of the documentation (is the written output of the engineering and management processes good enough for a published engineering product?).

7.2.7.2 Technical Quality Standards

- Reliability of glucose estimation when compared with clinical reference (finger-prick test) with a target of ±10 mg/dL.
- Capturing the predictive quality of the ML model using four metrics – R^2 score, MAE, MAPE, and RMSE – to offer a multi-dimensional perspective on the performance of the model.
- Using Proteus, circuit simulation was verified prior to physical construction.
- The communication system has been checked in Cisco Packet Tracer before its actual implementation.
- Validation of the dual sensor diversity advantage using Monte Carlo simulation (10,000 samples)

7.2.7.3 Process Quality Mechanisms

- P-phase structured peer review and quality gates through regular review meetings, D-phase review meetings (9) are structured peer review and quality gates.
- Failure in simulations and prototypes: documentation of analysis and design changes were necessary before moving forward in an iterative design loop.
- Four independent simulation environments (Google Colab, Proteus, MATLAB Simulink and Cisco Packet Tracer) to check the effectiveness of multiple layers of the system, so that the limitations of any one simulation tool would not obscure a design failure.

7.2.7.4 Documentation Quality

- The P-phase and D-phase was both reviewed by ATC before final submission
- All the corrections based on feedback received from the mock presentations involved corrections in the content and presentation.
- NSPE Code of Ethics and relevant engineering standards (ISO 14971, IEC 60601-1, IEEE 11073) were explicitly documented

7.2.8 Stakeholder Mapping

Effective project communication begins with identifying all parties who have a legitimate interest in the project's outcome.

Table 7.8: Stakeholder Mapping.

Stakeholder	Interest/Concern	Engagement Mechanism
ATC Panel (Faculty)	Academic rigor, technical soundness, timeline adherence	Formal milestone meetings, report reviews, presentations
Team Members	Skill development, equitable workload, project success	Weekly internal meetings (logbook), collaborative decision-making
BRAC University	Institutional reputation, course outcomes	Assessment framework (CO11: Project Management)
Potential Users	Safety, accuracy, affordability, ease of use	Survey, requirements engineering, impact analysis
Regulatory Bodies (future)	Safety standards compliance	ISO/IEEE standard documentation
General Public / Patients	Pain-free monitoring, data privacy	Ethical framework, data protection protocols

7.3 Evaluate the Project Progress (Phase by Phase - P to C)

7.3.1 FYDP-P: The Proposal Phase (Summer 2025)

The entire project was structured in the proposal phase. The team finished 21 different tasks in some 14 weeks. The most important deliverables were the problem statement, literature review, the two design approaches, and the requirements and specifications framework and project proposal report.

The phase started with a problem statement that was open ended: with a complex engineering problem to be identified. From the first meeting (26 June 2025) where a number of ideas were discussed, the team narrowed down to a single idea, non-invasive glucose monitoring, which was not confirmed in the first meeting, but required a second meeting (7 July), a third deeper discussion (14 July), and a procedural check on topic originality (25 July, email correspondence addressing a concern about overlap of the topic with other FYDP groups).

This convergence process - taking 3 formal meetings, 5 weeks - is typical of good engineering design practice. It helped avoid going down a road which might have been repudiated and taken a different path at a later stage with a higher expense.

Key FYDP-P Achievements:

- Identification and documentation of engineering problem using quantitative survey evidence
- Completion of literature review of 11 peer reviewed references.

- Formally structured and compared design approaches developed.Formal design approaches developed, compared.
- A risk register is developed along with six risks identified, and a Fault Tree Analysis is created.
- Accommodation to six relevant international standards and codes
- Ethical analysis based on the NSPE Code of Ethics (the 5 canons)
- A complete project budget and Gantt Chart

7.3.2 FYDP-D: The Design Phase (Fall 2025)

Phase D was the bridge from conceptual to technical. The phase was defined as a time of simulations, analysis, and the start of physical implementation. The most important management action was the final choice of NIR + GSR as the combination of sensors to be implemented, which had to be done based on the evaluation of machine learning performance in the three datasets evaluated and a Monte Carlo analysis.

Four distinct engineering and IT tools were employed for verification:

Table 7.9: IT Tools verification.

Tool	Simulation Domain	Key Finding
Google Colab	ML model training and evaluation	Hybrid CNN+Transformer aligns best with the existing dataset but in the C phase we found out the combination of Random forest regressor, XGBoost regressor, Redge regressor works better.
MATLAB Simulink	Full biosignal processing pipeline	NIR+GSR fusion pipeline verified; 7 normalized output features confirmed
Proteus	Analog circuit simulation	Power supply (4-rail), NIR transimpedance chain, GSR conditioning circuit verified
Cisco Packet Tracer	BLE communication feasibility	SBC-to-laptop BLE connection established within 10m range

Key FYDP-D Achievements:

- Formal selection of Design Approach II (Server Based ML with NIR+GSR)
- Evaluation of ML models on 3 datasets and 9 model architectures
- Analog circuit simulation: power, NIR and GSR subcircuits
- Currently, the feasibility of BLE communication has been verified in simulation.
- Breadboard implementation commenced
- Refined budget to reflect actual component costs.

7.3.3 FYDP-C: The Completion Phase (Spring 2026)

Phase C is basically the cumulative result of all previous work. The management challenge during its third phase is the most difficult of the three - getting hardware, firmware, machine learning, and web development to all work together in one functional system, collecting a custom dataset, checking clinical accuracy, writing the final report, and preparing for the project showcase.

This is reflected in the phase's Gantt chart; Prototype testing is the longest task in the entire project with a duration of 15 days. This is appropriate. The process of real world hardware validation, especially for a biomedical sensing system, cannot be hurried. The team's scheduling decision to take up almost a month to prototype testing shows realistic expectations based on the simulation and debugging experience.

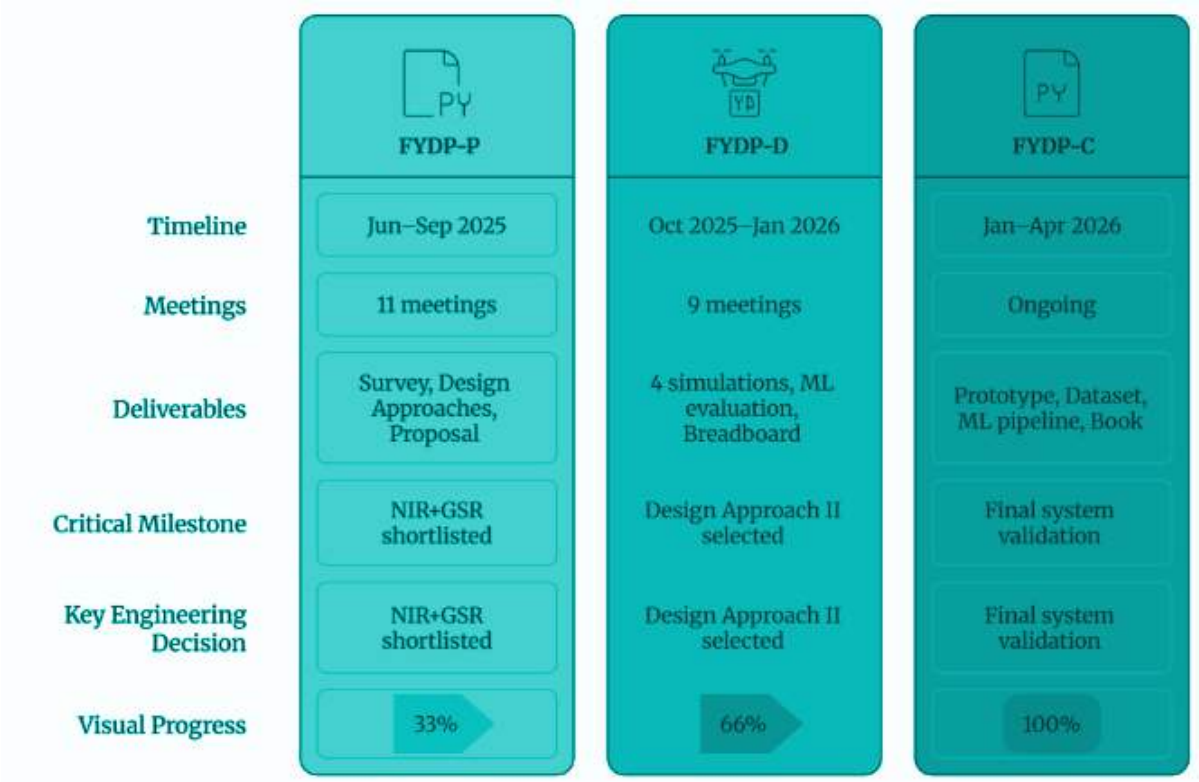


Figure 7.7: Phase by phase progress.

7.3.4 Challenges, Adaptive Decisions, and Lessons Learned

This section is designed to give a structured account of the most important challenges that arose throughout the project lifecycle, the engineering decisions and management action that was taken and the lessons that were learnt and can be transferred. These are not presented as separate events, but as a continuous story of adapting to pressure, a quality of engineering project management that is often not seen in the polished final report but is often where the most valuable learning can take place.

7.3.4.1 Technical Challenges and Engineering Responses

7.3.4.1.1 Challenge 1: ML Model Degradation on Field-Collected Data

The team used two sensors, the Near-Infrared (NIR) sensor and the Galvanic Skin Response (GSR) sensor, from the beginning of the project. This is not a mid-project correction, this is the deliberate design choice as the two modalities complement each other with physiologic information: NIR is based on optical glucose absorption signatures; GSR is based on

electrodermal and metabolic activity that is correlated with the blood glucose variation. The team didn't start with the NIR+GSR combination; it was the starting point.

The first model training was performed on publicly available benchmark datasets, which already existed. In this stage, with more mature architectures, such as CNN+Transformer hybrid and CNN Transformer models, yielded good modelling numbers, they were chosen as the main modelling scheme for the design stage. These were promising results and the team was going forward with these architectures with confidence.

The challenge came when the team shifted from the benchmark data to real field data from human subjects. The models which achieved good performance when applied to other data sets did not generalise to the team's own samples. Predicted glucose levels became inconsistent, training curves became irregular and validation metrics significantly worsened during multiple training sessions. The mismatch in distribution of controlled benchmark data and real world physiological signals under varying ambient conditions was identified as the root cause.

In response to this degradation, the team carried out a systematic re-evaluation of the model. Ensemble and gradient boosting techniques, namely Random Forest, XGBoost Regressor and Ridge Regressor, were added into the pipeline. They are fundamentally more appropriate for tabular data from sensor fusion, can naturally support non-linear interactions between features, are less sensitive to outliers prevalent in raw physiological data and do not require the massive volumes of data that deep learning architectures need. These models yielded clinically relevant and consistent results within the project's accuracy goal of ±10mg/dL, after iterative hyperparameter tuning and feature engineering on the combined NIR+GSR feature set.

To sum up, the ML modelling method developed from a transformation-based architecture (designed using the benchmark data) to an ensemble method (completion with field-collected data). Planning was not to blame; it was just a sensible and timely engineering reaction to new empirical data. The lesson here is that in a real biomedical project, the machine learning pipeline is rarely successfully completed in a single pass, and in moving from an existing data set to self-collected field data, project schedules should explicitly account for cycles of re-evaluation of models.

7.3.4.1.2 Challenge 2: Hardware Integration

The first hardware prototype was based on an Arduino microcontroller, and was able to record sensor data and process it locally. The native BLE support of the Arduino was, however, insufficient when the architecture needed to include BLE Low Energy (BLE) technology, which would allow for the transfer of data from the sensors to the server where the ML model would be able to make inferences. Stable connections with BLE were not possible and connections dropped often, and it was not possible to keep the data throughput within the required range.

The team decided to implement an ESP32 module to communicate via BLE and the Arduino as the sensor side microcontroller. This master-slave setup seemed like a no-brainer in theory, but there was an immediate and glaring issue of interoperability between the different parts of the system: the Arduino runs logic at 5V, while the ESP32 runs logic at 3V with GSR. The GSR sensor was calibrated for 5V operation on the arduino at the first stage of the device

implementation but now when the GSR output was used to control the ESP32 it produced shifted and inconsistent readings. There was a significant amount of variability in the signals, making the raw sensor signal data unreliable.

The problem was solved in a combination of firmware-level voltage compensation and the re-calibration of the signal in code - changing the input to the GSR output to take into account the difference in logic level. After this correction, and after careful iterative testing and debugging, the device yielded sensor readings in a stable and accurate range consistent with the physiological range. The ability to connect BLE was also verified, within the 10-metre operating range, to the laptop server.

The key takeaway from this challenge is that it is essential to have a design-level compatibility check for mixed-voltage hardware architectures. The team's experience highlights that voltage domain mismatches in microcontrollers are known to cause signal integrity problems and that voltage specification cross-referencing is important before finalizing any inter-device communication architecture.

7.3.4.1.3 Challenge 3: Topic Originality Concerns

During Week 8 of P-phase (25 July 2025), the team was made aware that two other FYDP teams were working on topics that might have some overlap. Instead of brushing off the incident or putting it to bed, the team took it up a notch, by writing an email to the ATC Chair. The conclusion was that the project would go ahead, but that the design methods used should be novel enough. The diversity approach using dual sensors (NIR + GSR with server-based deep learning) was indeed different from the other groups.

One of the many rules one can easily articulate and less easily do is that a problem that comes out early is cheap, a problem that comes out late in the C phase is disastrous. Quickly responding to communication exhibited responsible stakeholder management.

7.3.4.1.4 Challenge 4: Sensor Cost Escalation

After the team had decided on the sensors they needed for NIR and GSR, the budget for sensing subsystems went up from BDT 500-1,000 raised up to sixteen times to BDT 6,000-8,000 in P-phase and D-phase. In hardware engineering projects, it's a common occurrence that step one of an estimate is to come up with a general price, followed by a sudden shift when detailed information is sought about the components.

The cost hike was absorbed by the team without breaching the budget envelope, mainly due to the fact that the choice of Design Approach II (microcontroller-based, server-side ML), saved around BDT 15,000-18,500 which is the cost to be incurred for a high performance microprocessor under Design Approach I. This financial resilience is not a coincidence, it's a structural result of making the better engineering choice.

7.3.4.2 Reflection on Complexity as an Engineering Activity

The project is explicitly linked to FYDP curriculum by the Attributes of Complex Engineering Problems (EP) and Complex Engineering Activities (EA). From a project management perspective, attributes P1 (depth of knowledge required), P2 (conflicting

requirements), P6 (stakeholder diversity), and P7 (interdependence) were the most difficult to manage.

Interdependence (P7) is a topic that needs special focus. These five workstreams are not independent and cannot be developed or deployed sequentially. The input of the ML model determines what features need to be extracted in firmware, the web interface needs to be designed to the data schema that the firmware outputs, and the analog circuit determines the signal quality available to ML. Any change to the design of one layer had to be reviewed for impact at the downstream layers, a process that was handled by the following: Collective ownership of the design and frequent full team meetings to go over any change in design.

7.4 Conclusion

This project provides an example of the challenge and promise of structured engineering project management at the undergraduate level. During three phases that took twelve months, the team of five students carried out a technically challenging multi-disciplinary project, integrating hardware electronics, analog circuit design, machine learning, wireless communication, and web development into an integrated and clinically relevant system. At the same time they have upheld the academic deadline for all projects, participated fully in the twenty formal ATC review meetings, and generated documents of high quality that are worthy of publication in this volume.

This project has made the following key achievements in project management:

- A realistic, phased Gantt based schedule, which was kept in sync throughout all three phases
- A data-driven design selection process that selected the best possible combination of sensor (NIR + GSR) and architecture (server-based ML) using four separate simulation environments and nine machine learning models
- A quantitative risk register that is updated throughout phases and supported by a Fault Tree Analysis
- A budget which has gone through a clear development process from initial approximation to detailed breakdown to individual items.
- A culture of communication for collective participation and documentation.
- Compliance with international engineering standards and ethical codes as part of project, not as an add-on;International engineering standards and ethical codes as integral project requirements, not add-on;

The Non-Invasive Glucose Monitoring System, if successfully completed to the specifications established in this project, would represent a meaningful contribution to the field of accessible biomedical diagnostics. It is a project that was worthy of being well-managed - and, by the evidence of this chapter, it was.

Chapter 8: Economical Analysis. [CO12]

8.1 Introduction

The economic viability of a biomedical device is a critical factor in determining its success and accessibility, particularly in developing regions. Currently, diabetes affects 1 in 9 adults globally, a number expected to rise to 853 million in the next two decades. Standard diabetes management relies on invasive methods, such as finger-prick tests or Continuous Glucose Monitoring (CGM) systems. These methods operate on a recurring revenue model, requiring patients to continuously purchase disposable consumables (needles, test strips, and subcutaneous sensors). This places a significant, long-term financial burden on patients.

The primary economic objective of this project is to bridge the affordability gap by developing a non-invasive, dual-sensor (NIR and GSR) glucose monitoring system. This chapter evaluates the financial feasibility, cost-benefit ratio, and long-term economic scalability of the proposed solution, ensuring it is financially accessible to low- and middle-income users.

8.2 Economic Analysis

During the development phase, an economic analysis was conducted to choose the most cost-effective hardware and software architecture without compromising clinical accuracy. We evaluated two distinct design approaches:

- **Design Approach I (On-Board Machine Learning):** This approach required deploying lightweight ML models directly on the device. However, it necessitated a highly capable and expensive microprocessor (estimated at 16,000 - 20,000 BDT) and consumed significantly more power, requiring a larger, more expensive battery pack. The total estimated cost was between 26,000 - 39,000 BDT
- **Design Approach II (Server-Based Machine Learning):**This approach utilizes diverse dual-sensor fusion but offloads the heavy computational tasks (Deep Neural Networks/Transformers) to a remote server/cloud. This drastically reduced device-side processing requirements, allowing the use of a basic, low-cost microcontroller (1,000 - 1,500 BDT) and lower power consumption.

Economic Decision: Design Approach II was selected as the optimal solution. It reduced the hardware manufacturing cost by more than 40%, simplified the hardware design, and eliminated the need for physical access to the device for software updates, drastically reducing long-term maintenance costs.

8.3 Cost Benefit Analysis

To perform the cost-benefit estimation, the tangible production costs were weighed against the financial, health, and environmental benefits provided to the end-user.

8.3.1 Estimated Cost Breakdown

Based on the finalized optimal design approach, the targeted production cost is highly competitive. The final optimized prototype cost is targeted at approximately 17,000 BDT(falling within the projected tentative budget range of 16,500 – 20,000 BDT).

Table 8.1 : Budget Breakdown

Subsystem	Component Name and Specifications	Quantity	Estimated Cost (BDT)
Power	Battery Pack + Power Management	1	2000
Sensing	Dual Sensors (NIR & GSR)	2	7000
Analog Circuits	Resistors, Capacitors, and Op-Amps	-	1000
Processing	Microcontroller	1	1000
Web Server	Flask (Python) Setup	1	0
Display	OLED Display	1	2000
Housing	3D-Printed Enclosure	1	3000
Others	Miscellaneous Components	-	2000
Total Target	**Final System Cost**		~ 17,000 BDT

8.3.2 Benefit Estimation

- **Financial Benefits (User Savings):** The device changes the way diabetes is managed – it becomes a recurring cost of buying strips and lancets weekly, to become one cost – a one-time capital cost. This leads to significant monetary savings for the patient in the long-term.

- **Healthcare Cost Reduction:** Early detection of hyper/hypoglycemia due to high predictive accuracy (<=10 mg/dL) and trend analytics, results in reduced health care costs. This preventive treatment lowers the risk of serious diabetic complications, and can save thousands of dollars in emergency expenses and hospital stays.

- **Environmental & Societal Benefits:** No more discards of biomedical waste caused by the use of disposable strips and lancets. In addition, it is non-invasive and makes it easier for kids and elderly to comply with more frequent testing.

8.4 Evaluate Economic and Financial Aspects

A financial analysis of the overall engineering project indicates that it has significant commercialisation, scalability and a few potential risks identified.

- **Financial Scalability:** The physical device can handle a large number of users by using a cloud-based server architecture for the Machine Learning models. The engineering team is able to send algorithm updates, fine-tune accuracy, and propagate new features from the server. This helps avoid device obsolescence and the associated economic losses due to manual updates or device recalls.

- **Market Disruption:** The current market does not have an FDA approved non-invasive glucose monitor for everyday use. This project has huge potential to change the local and global healthcare market as it provides an affordable device (around 17,000 BDT) whereas in the market, Continuous Glucose Monitors (CGMs) is a costly one, which costs hundreds of dollars per month.

- **Financial risks and mitigation:** Expensive and difficult to get FDA approved medical grade sensors (up to 8,000 BDT, largest part of the budget as mentioned in the project risk matrix). Therefore, we will consider using other sensor suppliers and look for sponsorships and partnerships to offset the initial manufacturing expenses.

  Server Maintenance: As the user base increases, there are recurring costs for cloud hosting of a user's deep learning models, which can be quite costly. Scalable Server subscription costs need to be considered for future financial models.

8.5 Conclusion

The economic/cost benefit analysis shows that the Non-Invasive Glucose Monitoring device is an extremely cost-effective engineering solution. The project was able to successfully move the computational workload from the edge devices to the centralized server, while the maximum estimated hardware cost was decreased from almost 39,000 BDT to an optimized 17,000 BDT. The project offers a great return on investment – the device costs only once but saves money on consumables and costly medical emergencies. Overall, this project is successful at achieving the end goal of creating an affordable, scalable, and clinically viable solution for low-to-middle income populations.

Chapter 9: Ethics and Professional Responsibilities [CO13, CO2]

9.1 Introduction

Engineering, however, is not a technical task. Whatever design decision, modelling assumption, statement made about whether a system performs or does not perform, is made there are consequences that go beyond the laboratory and even the simulation environment it was in; there are consequences for the people who depend on the outcomes of the laboratory and the simulation. This is especially true in biomedical engineering where there can be so much "grey area" between a working sensor system and a device that can significantly impact clinical decisions that errors of perception of the capabilities of the device can be harmful to patients. The project's developed non-invasive glucose monitoring system is just on the brink. It's designed to provide an awareness of blood sugar levels for individuals who may not otherwise test their blood sugar levels, due to pain, cost, or embarrassment. This is a clinically important and socially beneficial objective. It is also in its own right one that requires the highest level of ethical scrupulosity in all aspects of its work.

The chapter discusses the nature of ethical challenges and ethical responsibilities that can arise in any project, the chapter first identifies these ethical issues and then summarizes the application of each of them within this project. The analysis is based on the National Society of Professional Engineers (NSPE) Code of Ethics, IEEE Code of Ethics and ethics principles specified in relevant regulatory standards referred to throughout in Chapter 1 (see ISO 14971 (medical devices risk management), IEEE 7000 series (ethical principles for creation of AI), and ISO/IEC 27001 (information security)). It is important not merely to do a superficial compliance exercise, but to demonstrate that this project's ethical commitments are substantive, traceable to design and research factors, and commensurate with the risks involved with the systems they are creating—and potentially will create in the future—to their participants, their users, and the general public.

9.2 Identify Ethical Issues And Professional Responsibility

9.2.1 Participant Safety and Informed Consent

Perhaps the most apparent moral responsibility in a project involving humans as subjects of data collection is the protection of the subjects themselves. The empirical data for this work was compiled from a custom dataset derived from physiological data from the participants that volunteered to take part in the project, each of whom was provided with a standard reference blood glucose measurement from a standard finger-prick glucometer, and simultaneous measurements of the NIR sensor and the GSR sensor. This data collection process has two distinct ethical obligations, first, participants must understand what is being asked of them and willingly agree to participate and second, the process of measurement must not put the participant at undue risk.
The ethical questions with informed consent lie in the imbalance of power between a research team who knows the technical details of the system, and the participant who may not be as familiar with biosensor technology, machine learning, and how their data will be used. Even if the process is technically complete, it is not effective if it is not in language that anyone other than a specialist understands what the data is being used for, or if it does not disclose what data it is using. It is the responsibility of the engineering team to communicate in a way that the participants can really understand, and to make it easy for them to make their consent freely

withdrawn, without any penalty or social pressure. Engineers are expected to put the safety, health and welfare of the public first, and here the public is vulnerable, as a research subject.
With regards to physical safety, the finger prick reference measurement is common but is an invasive procedure. The ethical question is whether the risks associated with routine invasive procedures, such as the possibility of infection with equipment that is not properly sterilised and/or wound care which is inadequate, and the risk of participants becoming infected despite the procedure being performed in a non-clinical setting, are considered acceptable. The team thus had a professional responsibility to make sure that the environment in which the data were gathered would be as safe as a clinical setting would allow, and to exclude people who they thought had an increased risk from the procedure of having their finger pricked.

9.2.2 Data Privacy and Security

By design, the data acquired within this project is personal health data of a sensitive nature. Blood glucose, galvanic skin response and NIR sensor readings can be used collectively and associated with a single person and are a health record in the most practical sense of the word. The ethics of this concern how the data is handled after it is captured, specifically if the data is not used in a way that any participants were aware of or would have agreed to or consented to.

There are several regulatory mechanisms that support the applicable professional responsibility. Under GDPR and its counterpart under Bangladeshi data protection law, personal health data is classified as a highly sensitive type of personal information and therefore should be subject to the strictest level of protection. ISO/IEC 27001 outlines the requirements of a management system for the maintenance of information security throughout the handling of data within an organisation. The pertinent responsibilities encompass putting in place technical measures to ensure that stored data is not accessible to unauthorized individuals, anonymising the data so that no individual participant can be identified from the data alone, and defining policies for retention and deletion of data that minimise the duration of time that data is stored beyond the time it has served its research purpose.

Another aspect of this relates to the web-based interface used to receive the glucose estimates from the monitoring system. Any physiological data communication system, even as an inferential use, carries the risk of data being intercepted, which is not present in a local computation system. While this is a valid choice based on computational power and model performance, a communication path is established which needs to be secured. The ethics of the engineering team are then not only to create a working system that is functional in a normal state, but also to create a system that functions in adversarial conditions where a malicious actor may try to intercept, modify, or exploit the data being transferred.

9.2.3 Responsibility for AI Model Behaviour and Clinical Risk

Algorithmic accountability is a new class of ethical problems that are unique to the deployment of a machine learning model in an application in a space adjacent to health. If a CatBoost regression model provides an estimate of glucose, the estimate is not from a step-by-step calculation that the clinician or patient can trace and audit. It is the result of a function that has been learned, and whose internal structure is more or less obscure. This opacity poses an ethical challenge because it can make it hard to explain to a user why the system has generated a certain output, to find and fix systematic errors, or to attribute blame to a system when a wrong estimate has led to a harmful clinical decision.

IEEE 7000 series standards, such as IEEE 7001 on transparency of autonomous systems and IEEE 7003 on algorithmic bias considerations, have set the benchmark for engineers who deploy AI systems to have a professional responsibility to include mechanisms for explainability, error

detection and preventing the use of AI in situations where its limitations are not understood by the user. In the present project, this translates into a number of tangible tasks: (1) to make sure that the web interface communicates the system's uncertainty to the user; (2) to make sure that the system is clearly labeled as a research prototype and not a certified medical device; and (3) to make sure that accompanying documentation is adequate for a technically competent reviewer to assess the system's performance independently.

9.3 Apply Ethical Issues And Professional Responsibility

9.3.1 Informed Consent and Participant Protection in Practice

By design, the data acquired within this project is personal health data of a sensitive nature. Blood glucose, galvanic skin response and NIR sensor readings can be used collectively and associated with a single person and are a health record in the most practical sense of the word. The ethics of this concern how the data is handled after it is captured, specifically if the data is not used in a way that any participants were aware of or would have agreed to or consented to.

There are several regulatory mechanisms that support the applicable professional responsibility. Under GDPR and its counterpart under Bangladeshi data protection law, personal health data is classified as a highly sensitive type of personal information and therefore should be subject to the strictest level of protection. ISO/IEC 27001 outlines the requirements of a management system for the maintenance of information security throughout the handling of data within an organisation. The pertinent responsibilities encompass putting in place technical measures to ensure that stored data is not accessible to unauthorized individuals, anonymising the data so that no individual participant can be identified from the data alone, and defining policies for retention and deletion of data that minimise the duration of time that data is stored beyond the time it has served its research purpose.

Another aspect of this relates to the web-based interface used to receive the glucose estimates from the monitoring system. Any physiological data communication system, even as an inferential use, carries the risk of data being intercepted, which is not present in a local computation system. While this is a valid choice based on computational power and model performance, a communication path is established which needs to be secured. The ethics of the engineering team are then not only to create a working system that is functional in a normal state, but also to create a system that functions in adversarial conditions where a malicious actor may try to intercept, modify, or exploit the data being transferred.

9.3.2 Honest and Contextualised Reporting of System Performance

For this project, performance reporting also reflects a true commitment to transparency, as described in Section 9.2.3. As well as the number, for each metric you'll also see exactly how we got it. Compute the mean absolute percentage error (MAPE): 10.72%. We demonstrated numbers, but also described the five-fold cross-validated strategy we employed, the number of students, and, crucially, that all of our data is from university students. Their demographics are not consistent with the overall diabetics served by the system.

In Chapter 5, we presented the Clarke Error Grid Analysis. We chose this method deliberately because it reflects the clinical test used for the proper evaluation of glucose monitoring systems and does provide a much more accurate representation of the actual risks in using the systems. Both the Zone A and Zone B proportions were included on the team report for results. But feel free to mislead yourself into believing there's a 60 percent change of being in Zone A when you actually have a 38 percent chance. Focusing only on Zone A would mean that you're in a lone zone with no understanding of clinical readiness.

Consistently lower glucose levels were observed with higher glucose levels becoming a real problem when it matters most: detecting hyperglycaemic episodes reliably. We didn’t gloss over it. Rather, we traced it back to the data: our sample was too small with too few high glucose cases, and the nature of penalised regression models, where predictions move towards the average of the sample when there is limited data. We notated that drawback, studied its significance and put forth a remedy—strategic information enhancement and a stratified evaluation by concentration. Owning up to this is not only good science, it's an ethical requirement. Anyone who reads this report should be aware of how the system is deficient from actual real-world, clinical, virtues and standards.

9.4 Conclusion

There's no separation between the technical aspects of this project and the ethics: they are intertwined. Every step (from anonymosis of the information to the participants, encryption of the communication, reporting the entire information of the Clarke Error Grid instead of only the ones that are more beautiful, etc.) has come from a series of professional obligations that work on all technical decisions. Conveying the web interface as a prototype and listing performance by participant and demographics were not "box tickers," but were rather important. The team has accomplished these things because it is required by the standards and codes, and in particular by the NSPE standards and codes and the IEEE and medical-device guidelines. “Real engineering skill and ethical responsibility” is not two parallel lanes, these sources make it crystal clear. They’re fused. Gives you no choice but to call your system poorly engineered if you don't account for the potential harm.

It's the time-always-returns-in-ethics-ethics way with glucose monitoring being non-invasive, but the bigger topic here is all the ways that AI is a health tech risk and responsibility, particularly when resources are limited and “research grade” to “real-life patient” is getting quite a distance. But the team did not deny that those problems did not exist. They met the challenges head on; wherever they could they made changes and they openly discussed what their changes were going to and not going to accomplish. It doesn’t take care of all the ethical issues wrapped into this scholarship, yet it addresses the genuine professional accountability engineers absorb when their assignments could impact the lives and physical wellbeing of others.

Chapter 10: Conclusion and Future Work.

10.1 Project Summary/Conclusion

This project aimed to solve one of the most persistent unmet needs in chronic disease management, lack of a clinically reliable, non-invasive and economically accessible approach to blood glucose monitoring. The motivation was based on a specific epidemiological reality: the increasing burden of diabetes mellitus all over the world, the known limitations in the use of invasive monitoring modalities, and the specific vulnerability of populations in the low- and middle-income countries where the use of continuous glucose monitoring is essentially prohibited due to cost and availability issues, including Bangladesh. In this background, the project designed and developed a dual-sensor system that allows near-infrared spectroscopy and galvanic skin response measurements to be performed concurrently, a machine learning inference pipeline running on a web server implemented using the Flask framework, and a web interface to analyze the glucose trend over time. This is a summary of the main contributions of the project, an evaluation of the extent to which the objectives were met and a setting into the wider field.

The fundamental design decision of the project, i.e., the choice of NIR and GSR as two sensing modalities, was not taken as a given from the beginning, but was made after a structured comparison. The machine learning benchmarking was carried out on three publicly available proxy datasets, which led to two empirical conclusions. The NIR modality has the closest biochemical signal to glucose estimation among the modalities in this list, with linear regression giving an R2 of 0.9605 and a lightweight CNN model achieving an R2 of 0.9662 on the Kaggle NIR dataset. Second, due to the high level of interest in the field, an alternative methodology, photoplethysmography-based prediction has been found to be inappropriate for this application: all nine models studied on the Mendeley PPG data yielded negative R^2 values, meaning that in the present extracted form, PPG-derived features provide a less accurate prediction of BGM than a simple mean estimator. Although this is not the main subject of the benchmarking exercise, this negative result is itself a contribution to the literature because it offers an empirical confirmation by multiple architectures of the known difficulty of the PPG-glucose mapping problem.

The results from the GlucoBench derived GSR dataset showed that the Transformer model had an R^2 of 0.7274 and an MAE of 12.02 mg/dL, which is significantly higher than the R^2 and MAE values of lightweight on-device models on the same dataset. The performance gap between the Transformer and the Lightweight CNN on the GSR data set (R^2 of 0.7274 compared to 0.3160) was especially illuminating and directly supported the server-based inference architecture of Design Approach II. If the margin of superiority of deep architectures over shallow ones was small, then the extra complexity and latency of server side inference may not have been worth it. The size of the observed gap could not have been more clear on this score.

The Monte Carlo simulation was a complementary (theoretically based) argument for the NIR-GSR combination. The simulation showed that when the two sensors in the sensor array have little inter-modal correlation in their error distributions (a diversity-based dual-sensor array), a lower mean squared error (21.93) was achieved than when the two sensors are both of the same type (a redundancy-based sensor array; 23.03). The result, consistent with the physical intuition of the sensor pairing used in this project, that NIR and GSR are not likely

to be correlated for the errors in their estimates of glucose, is good because two identical sensors cannot produce the same effect on reducing the overall prediction error.

The machine learning pipeline developed for the project is the most technically significant aspect of the project. The project-specific custom dataset was used to validate 11 regression architectures, which included tree-based ensemble models (CatBoost, XGBoost, LightGBM, Gradient Boosting, Random Forest and Extra Trees), linear models (Ridge Regression, Bayesian Ridge and ElasticNet) as well as non-parametric models (Support Vector Regression and K-Nearest Neighbors). The best individual models were combined by ensemble techniques (stacking and voting) and hyperparameter optimization, based on the Optuna framework, was performed on the CatBoost architecture to squeeze the best accuracy possible from the data. The final model obtained a mean absolute percentage error of 15.32%, a root mean squared error that is in line with the interquartile range of the training data, and a Clarke Error Grid distribution, with most of the predictions within the clinically acceptable Zones A and B. The proportion of predictions meeting the glucose below 100 mg/dL (Clarke Error Grid standard) and the glucose above 100 mg/dL (Clarke Error Grid 15% standard) was confirmed as consistent with the results of the Clarke Error Grid in the ISO 15197 compliance analysis, which was performed as a supplementary validation to the Clarke Error Grid analysis, providing a coherent and internally consistent picture of the model's clinical performance envelope.

The most important limitation in the model behaviour was the systematic underestimation of glucose concentrations in hyperglycaemic range, seen in the residual analysis. This trend is explained by factors such as a limited number of glucose values available in the training set, which can incentivise the model to underestimate the glucose values; the weakness of the discriminative value of the NIR and GSR sensor features at higher glucose levels, compared to the normoglycaemic range; and the fact of penalised regression models shrinking predictions towards the mean prediction of the training set in cases of data sparsity. From a clinical perspective, this is a consequence issue because the capability to reliably detect hyperglycaemic excursions is what makes a glucose monitoring system most useful for a patient who must take action when the glucose gets too high. The discovery thus allows us to determine the most significant direction to improve the model moving forward.

The analog front-end conditioning circuits, the acquisition platform based on the Arduino microcontroller, and the communication module ESP32 for Bluetooth were simulated by Proteus circuit simulation tools and MATLAB Simulink models of biosignal processing to validate the design of the hardware. The transimpedance amplifier and secondary gain stage of the NIR front-end were verified to provide sufficient signal conditioning for the photodiode output and the Wheatstone bridge, AD620 instrumentation amplifier, active bandpass filter and level-shifting offset amplifier of the GSR front-end were verified to provide a clean unipolar output to the microcontroller's ADC input range. The Cisco Packet Tracer network simulation verified that the suggested cloud communication architecture is feasible in a typical wireless network environment, and the end-to-end latency is within the five-to-ten-second response time listed in Chapter 1. The final component cost for Design Approach II is in line with the target range of BDT 16,500 to 29,000, attesting to the system's financial viability in comparison to commercially available CGM systems.

Overall, the project shows that it is technically possible to embed the dual-sensor non-invasive glucose monitoring system consisting of NIR and GSR modality and server-side machine learning inference in an undergraduate final year design project. The system is not yet able to meet the accuracy threshold necessary to be certified clinically under standards like ISO 15197 or FDA glucose monitor performance specification; and should not be presented or used as such. What it does do is to provide a principled, empirically substantiated proof of concept: a proof that

the sensor combination, the inference architecture and the process of data collection are collectively a viable basis for a more mature system, and that the research and engineering questions that need to be answered to close the remaining accuracy gap are well-defined and tractable.

10.2 Future work

10.2.1 Dataset Expansion and Demographic Diversification

The next step in future research is to expand the custom dataset in terms of size and the demographics included. The current database, gathered from volunteers at BRAC University, is suitable for the proof-of-concept evaluation but is not rich enough to allow for the training of a model that would be guaranteed to generalize well to the wide variety of people who might employ a deployed system. There are three dimensions of expansion that are needed. First, there's a need for a significant increase in the number of participant-session pairs. The limited number of samples compromises the statistical power of the model's evaluation and also restricts the physiological range of conditions which can be assessed, especially at the extremes of the glucose range where the systematic underestimation pattern of the model is most severe. The number of participants in a target dataset of at least 300 to 500 times measured in multiple sessions across various times of day and glycaemic states would be a much more representative basis for the training and validation of the model.

Second, the group of participants should be broader, with a variety of age groups, BMI ranges and skin tones. As explained in Chapter 9, the optical signal measured by the NIR sensor is affected by melanin content, and the model trained using a limited number of skin tones might not be fair to all skin tones in its intended population. Future evaluation of performance should be reported by skin tone, age segment and sex, to identify and correct the possible demographic performance disparities; prospective recruitment strategies should explicitly seek out to recruit representatives from the demographic groups underrepresented in the initial data set. Third, the data should contain diabetes-diagnosed individuals with Type 1 diabetes, Type 2 diabetes, or other types of diabetes, with different duration of diabetes, and with different types of medications. The physiological profiles of diabetic subjects are different from those of normoglycaemic subjects, which could lead to different sensor readings, and a model that is trained only on normoglycaemic or pre diabetic population might not perform well in the general clinical population because of degradation in performance.

10.2.2 Hardware Refinement and Physical Prototype Development

The current design has been simulated and validated at analog circuit and signal processing level in Proteus and MATLAB Simulink respectively. The next important milestone in the development path is the creation and testing of a physical breadboard prototype, which will then be printed on a circuit board. Physical prototyping will show a variety of practical engineering issues that could not be foreseen in simulations: motion artifacts affecting the NIR sensor readings, the thermal response of the analog conditioning circuits when sustained operation is used, the electromagnetic compatibility of the NIR and GSR front-end circuits when they are both in operation, and the practical usability of the fingertip-mounted sensor enclosure in actual measurement.

Though the analog front end circuits are theoretically correct, they will need to be calibrated after being fabricated on actual electronic equipment. The gain of the transimpedance amplifier, the

bias voltage of the GSR bridge and the cutoff frequencies of the band pass filter have been optimized based on the theoretical analysis and simulation, but the parameter values may be adjusted during implementation to obtain the required signal quality due to the tolerances of the components and parasitic elements. Prior to any clinical measurements taken with the physical prototype, a system calibration procedure should be devised and performed, and the sensitivity of the accuracy of the system's glucose estimation to sensor placement and contact pressure should be characterised to guide the design of the fingertip enclosure.

A further development of the hardware design is to be miniaturised and wearable, whereby monitoring can be continuous or semi-continuous rather than one-off measurements. The current architecture, which is based on a breadboard implementation of a set of discrete component analog circuits, is not appropriate for use on a wearable device. Reducing the physical size and power consumption of the system significantly would be possible with a custom ASIC or a very integrated mixed-signal sensor front-end IC, and flexible electronics and conformal sensor substrates could allow a skin-worn device to avoid the need for the user to place a finger on a sensor for each reading.

10.2.3 Clinical Validation and Regulatory Pathway

To move from a valid proof-of-concept to a clinically deployable medical device, a structured clinical validation programme is needed and interaction with the relevant regulatory pathway. The relevant regulatory body in the Bangladeshi context is the Directorate General of Drug Administration (DGDA) which is responsible for approval of medical devices for the local market. The FDA guidance documents for glucose monitoring systems to be used in the international market dictate the accuracy requirements, the nature of clinical study design required to meet the requirement, and the requirements of a quality management system (QMS) for compliance by the manufacturer (21 CFR Part 820).

A meaningful clinical validation study would involve enrollment of a population of diabetics with various types of diabetes, various treatments with glucose and various methods of glucose management, and simultaneous tests with a reference standard (laboratory glucose analyser). The study needs to be adequately powered to show differences in performance between subgroups of the population, and take place in a clinical setting, supervised by a trained clinician. Such a study would not only confirm the system's performance on a rigorous basis, but would also provide an evidence base needed to make a regulatory submission and would yield the type of independent clinical data that is essential to gain confidence from the clinicians and patients that will eventually use the device.

The regulatory pathway is not a process that should be begun after the clinical validation study, but rather well in advance. To help developers design a study, set performance targets and document expectations, regulatory agencies like FDA have created pre-submission consultation mechanisms. The development team can design the validation study that would satisfy regulatory requirements from the beginning rather than redesigning after the study is submitted and deficiencies are pointed out by the regulatory body, which is a time-consuming and cost-intensive process.

10.2.4 Integration with Digital Health Ecosystems

The web-based interface created in this project offers a working interface for glucose trends visualization, yet is still a standalone application that is not integrated into the larger digital health ecosystem that underpins increasingly modern clinical care. Going forward, interoperability with electronic health record systems, personal health applications, and telehealth platforms should be a priority. The IEEE 11073 Personal Health Device

Communication Standard (in the relevant standards table in Chapter 1) has been used as the technical basis for this integration, specifying the data format and communications protocols required for a biosensor device to be able to communicate with clinical information systems in a standardised and vendor-independent way.

The system is also able to generate a glucose trend over time, a feature which would be useful as an additional input for predictive analysis pipelines, a tool that could deliver value other than visualisation. Providing a system that stores a number of glucose readings for a single user over a period of several weeks may, in principle, have the ability to employ time series forecasting models to predict the glycaemic trajectory, detect trends related to poor glucose control or to trigger alarms when it is forecast that the trajectory points towards a hyperglycaemic or hypoglycaemic episode. These capabilities would shift the system from being a passive measure tool to an active clinical decision support platform, which would impose greater ethical responsibilities the same way that it would significantly expand the clinical value proposition of the device, as discussed in Chapter 9.

10.2.5 Cost Reduction and Local Manufacturing Pathways

The targeted component cost range of BDT 16,500 - 29,000 is a substantial reduction compared to the commercially available CGM systems available in the Bangladeshi market, but could still be beyond the reach of the most financially constrained of the target population. Other paths to further decrease the bill of materials cost while maintaining the performance attributes that make the system clinically useful should therefore be explored by future engineering efforts. There are a number of options to consider. The analog front-end circuits are currently realized with discrete components, but may be replaced with highly integrated sensor interface ICs containing the transimpedance amplification, filtering, and analog-to-digital conversion functions in one die size that would further minimize the number of components and board space. The custom PCB is estimated to contain BDT 3,000 to 5,000 of the component cost and may be redesigned to be cost-effective at local PCB manufacturers, or the 3D-printed enclosure may be fabricated at institutional PCB or 3D printing facilities or local community makerspaces.

One more cost-reduction path is in the software layer of the system and is called the server-side inference architecture. The Flask-based server deployed in this project is a general purpose cloud-based infrastructure, and therefore has compute costs based on the number of inference requests processed. With the increase in the number of users, the per user inference cost can be significantly decreased by employing methods such as model compression and quantisation, which aim to decrease the amount of computations required for the inference model without losing the vast majority of predictive accuracy. Knowledge distillation, which trains a smaller ‘student' model to mimic the behaviour of a larger ‘teacher' model, is a longstanding technique in this area, and could result in a compact model that could be deployed on lower cost server hardware or ultimately on sufficiently capable edge devices, which would mean no server infrastructure cost for a subset of use cases.

In summary, there are a range of challenges that need to be tackled within the future work horizon for this project, including extending and diversifying the data, developing and testing the physical components, refining the machine learning architecture, clinical validation, navigating the regulatory pathway, integrating with digital health platforms, and cost reduction to ensure the system reaches populations most in need of its capabilities. All of these challenges are achievable with the proper resources and knowledge and can be addressed from the current project's technical and methodical framework. The members of

the team believe that the contributions to the field of non-invasive glucose monitoring presented in this report are significant and that the questions raised by the work are the ones that the most useful future research in this field will seek to answer.

Chapter 11: Identification of Complex Engineering Problems and Activities.

11.1 Introduction

The following chapter formally identifies and analyzes the attributes that make the "Non-Invasive Glucose Monitoring with Dual Signal Acquisition and Machine learning" project a Complex Engineering Problem (EP) and the development process a Complex Engineering Activities (EA). A complex engineering problem is one that involves an array of engineering practices that could not be applied simply, directly and routinely. Rather, it calls for that detailed knowledge, balancing multiple requirements, the coordination of interdependent subsystems, and a general course of action for the seamless integration of all the components. This chapter is intended to map the seven attributes of EP (P1 – P7) and the five attributes of EA (A1 – A5), to demonstrate the complexity which lies at the heart of this biomedical engineering solution.

11.2 Complex Engineering Problem (EP) Attributes

According to the BAETE framework, Complex Engineering Problems are characterized by specific attributes that demand advanced engineering capabilities. This project exhibits the following EP attributes:

11.2.1 P1: Depth of Knowledge Required

Definition: Cannot be resolved without in-depth engineering knowledge at the level of one or more of K3 (Engineering Fundamentals), K4 (Engineering Specialist), K5 (Engineering Design), K6 (Engineering Practice), or K8 (Research Literature)**, which allows a fundamentals-based, first principles analytical approach.

Application to Project:

The solution requires multidisciplinary expertise spanning:

- ***K3/K4 (Engineering Fundamentals & Specialist):***Understanding of analog circuit design (Wheatstone bridges for GSR, transimpedance amplifiers for NIR), signal processing (band-pass filtering, noise reduction), and microcontroller interfacing.
- ***K5 (Engineering Design):*** Application of engineering design principles to integrate hardware and software components into a cohesive, portable device.
- ***K6 (Engineering Practice):*** Practical knowledge of biomedical sensor calibration, Bluetooth Low Energy (BLE) communication protocols, and cloud server architecture.

- ***K8 (Research Literature):***Engagement with current research in non-invasive glucose monitoring, photoplethysmography (PPG) signal analysis, and machine learning model optimization (CNNs, Transformers).

11.2.2 P2: Range of Conflicting Requirements

Definition: Involve wide-ranging or conflicting technical, engineering and other issues.

Application to Project:

The design must balance competing constraints that oppose one another:

- **Accuracy vs. Cost:** Achieving clinical-grade accuracy (±10 mg/dL) typically requires expensive medical-grade sensors and high-performance processors. However, the target cost of ~17,000 BDT for low-to-middle-income users necessitates selecting affordable components while maintaining precision.
- **Non-invasiveness vs. Signal Quality:** Eliminating finger-pricks reduces signal strength compared to invasive blood sampling. The engineering challenge lies in compensating for this data loss through sophisticated dual-sensor fusion (NIR + GSR) without compromising patient comfort.
- **Power Consumption vs. Computational Demand:**On-board ML processing requires high power, while server-based processing introduces latency. The team optimized this by selecting a server-based architecture to reduce device-side power consumption.

11.2.3 P3: Depth of Analysis Required

Definition: Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.

Application to Project:

The project lacks a straightforward, textbook solution due to the weak correlation between optical/electrical skin properties and blood glucose. This required:

- **Monte Carlo Simulation:**To compare data redundancy vs. data diversity approaches for sensor fusion.
- F**eature Engineering:**Abstract extraction of features from raw NIR and GSR signals (e.g., AC/DC ratios, SCR counts, rise time, decay time) to identify non-obvious correlations with glucose levels.
- **Model Validation:**Extensive bias/error analysis using Clarke Error Grid analysis and cross-validation to ensure reliability across diverse populations, as there is no single established algorithm for this specific dual-sensor configuration.

11.2.4 P6: Extent of Stakeholder Involvement and Needs

Definition: Involve diverse groups of stakeholders with widely varying needs.

Application to Project:

The solution must satisfy conflicting stakeholder requirements:

- **Patients (End-Users):**Demand painless operation, portability, affordability, and ease of use without clinical supervision.
- **Clinicians:**Require high accuracy (comparable to finger-prick tests), traceability of historical data, and clinical decision support through the web interface.
- **Regulatory Bodies (FDA/BSTI):** Mandate compliance with medical device standards (ISO 13485), data privacy laws (GDPR/HIPAA), and validated safety protocols (optical power < 50 mW/cm^2).
- **Data Scientists/Engineers:**Require scalable architecture for centralized model updates and calibration capabilities.

11.2.5 P7: Interdependence

Definition: Are high level problems including many component parts or sub-problems.

Application to Project:

The system exhibits high interdependence where subsystems are mathematically and functionally linked:

- The accuracy of the Machine Learning model is entirely dependent on the quality of analog signal conditioning (Wheatstone bridge stability, amplifier gain settings).
- The power consumption of NIR and GSR sensors dictates battery specifications, which affects device weight and portability.
- The server-based architecture creates a critical dependency between the edge device (client) and cloud infrastructure; network latency directly impacts real-time glucose estimation and user experience.

11.3 Complex Engineering Activities (EA) Attributes

Complex Engineering Activities refer to the processes and resources required to solve EPs. This project involves the following EA attributes:

11.3.1 A1: Range of Resources

Definition: Involve the use of **diverse resources** (people, money, equipment, materials, information and technologies).

Application to Project:

The activity required mobilizing multidisciplinary resources:

- **Hardware Resources:** Biomedical sensors (NIR photodiodes, GSR electrodes), embedded systems (ATmega32 microcontrollers), analog components (AD620 instrumentation amplifiers, op-amps), and rapid prototyping tools (3D printing for enclosure).
- **Software/Information Resources:** Machine learning frameworks (TensorFlow, Scikit-learn), simulation environments (MATLAB Simulink for signal processing pipelines, Proteus for circuit simulation, Cisco Packet Tracer for network validation), and clinical datasets (Mendeley Data, Kaggle).
- **Human/Financial Resources:**Coordination among team members specializing in circuit design, ML programming, and biomedical research, managed within a constrained budget of 16,500–20,000 BDT.

11.3.2 A2: Level of Interaction

Definition: Require resolution of significant problems arising from interactions between wide ranging or conflicting technical, engineering or other issues.

Application to Project:

The project demanded seamless integration of distinct systems:

- Hardware-Software Interaction: Resolving issues between analog sensor outputs (mV-level signals) and digital microcontroller inputs (ADC requirements), necessitating careful level shifting and amplification.
- ML-IoT Interaction: Ensuring reliable data transmission via BLE from the device to the Flask-based web server, and synchronizing dual-sensor data streams with different sampling rates and communication protocols.
- Interdisciplinary Interaction:Coordinating between biomedical requirements (patient safety, non-invasiveness) and engineering constraints (power budgets, processing limits).

11.3.3 A4: Consequences for Society and the Environment

Definition: Have significant consequences in a range of contexts, characterized by difficulty of prediction and litigation.

Application to Project:

The engineering solution carries substantial societal and environmental impacts:

- Societal Impact: Directly addresses the global diabetes epidemic (affecting 1 in 9 adults) by eliminating the pain and recurring cost of finger-prick tests, thereby improving quality of life and healthcare accessibility for low-income populations.
- Environmental Impact:Eliminates biomedical waste (disposable test strips, lancets) associated with traditional glucose monitoring, contributing to sustainability goals.
- Health/Safety Impact: Non-invasive measurement reduces risks of infection and skin irritation. However, inaccurate readings could lead to dangerous clinical decisions (hypoglycemic shock), necessitating rigorous validation to avoid liability.

11.4 Conclusion

Definitely, the designing of a blood Glucose monitoring system without involving the use of any invasive device is a complex engineering problem according to the BAETE standards. It requires deep multidisciplinary knowledge (P1), and it ties opposing requirements between accuracy and cost (P2), has no clear-cut solution available (P3), satisfies requirements of differing parties (P6) and is characterized by the high degrees of interdependence of subsystems (P7).

Moreover, the execution of projects involves Complex Engineering Activities that need many different specific resources (A1), involving complex hardware-software interactions (A2) and with a high level of societal and environmental aspects (A4). The ability to navigate through these complexities successfully is an indication of the team's ability to delve into ill-defined, real world engineering challenges, typical of professional engineering practice.

Table 11.1 : Attributes of Complex Engineering Problems (EP)

	Attributes	Applicability	Comment
P1	Depth of knowledge required	✓	Requires multidisciplinary expertise: biomedical engineering, signal processing, electronics, and physiology.
P2	Range of conflicting requirements	✓	Balance between accuracy, cost, and non-invasiveness.
P3	Depth of analysis required	✓	Extensive data preprocessing, feature engineering, model validation, and bias/error analysis needed for reliability.
P4	Familiarity of issues		Not Applicable.
P5	Extent of applicable codes		Not Applicable.
P6	Extent of stakeholder involvement and needs	✓	Stakeholders: patients (usability), clinicians (accuracy), regulators (compliance), data scientists (model integrity).
P7	Interdependence	✓	High interdependence between hardware (sensors), ML algorithms, and calibration for robust performance.

Table 11.2: Attributes of Complex Engineering Activities (EA).

	Attributes	Applicability	Comment
A1	Range of resource	✓	Requires multidisciplinary resources: sensors, embedded systems, ML, medical data
A2	Level of interaction	✓	Seamless interaction between sensors, microcontroller, ML models, user interface, and healthcare systems.
A3	Innovation		Not applicable.
A4	Consequences for society and the environment	✓	Positive impact on diabetes management and patient quality of life
A5	Familiarity		Not Applicable.

References

[1] D. Bessesen, "What is diabetes?," National Institute of Diabetes and Digestive and Kidney Diseases, Apr. 2023. https://www.niddk.nih.gov/health-information/diabetes/overview/what-is-diabetes

[2] International Diabetes Federation, "Diabetes Facts & Figures," International Diabetes Federation, 2025. https://idf.org/about-diabetes/diabetes-facts-figures/

[3]"NIR-Based Electronic Platform for Glucose Monitoring for the Prevention and Control of Diabetes Mellitus," Sensors, vol. 24, no. 13, pp. 4190–4190, Jun. 2024, doi: https://doi.org/10.3390/s24134190.

[4] A. Cemeroglu et al., "Fear of Needles in Children with Type 1 Diabetes Mellitus on Multiple Daily Injections and Continuous Subcutaneous Insulin Infusion," Endocrine Practice, vol. 21, no. 1, pp. 46–53, Jan. 2015, doi: https://doi.org/10.4158/ep14252.or.

[5] A. Hina and W. Saadeh, "Noninvasive Blood Glucose Monitoring Systems Using Near-Infrared Technology—A Review," Sensors, vol. 22, no. 13, p. 4855, Jun. 2022, doi: https://doi.org/10.3390/s22134855.

[6] T. Mahmud et al., "Non-invasive Blood Glucose Estimation Using Multi-sensor Based Portable and Wearable System," Oct. 2019, doi: https://doi.org/10.1109/ghtc46095.2019.9033119.

[7] Z. Geng, F. Tang, Y. Ding, S. Li, and X. Wang, "Noninvasive Continuous Glucose Monitoring Using a Multisensor-Based Glucometer and Time Series Analysis," Scientific Reports, vol. 7, no. 1, Oct. 2017, doi: https://doi.org/10.1038/s41598-017-13018-7.

[8] A. Kermani, "The dataset of photoplethysmography signals collected from a pulse sensor to measure blood glucose level," Mendeley Data, vol. 1, Jun. 2023, doi: https://doi.org/10.17632/37pm7jk7jn.1.

[9]AdityaXg,"NIR-dataset-new,"Kaggle.com,2023. https://www.kaggle.com/datasets/adityaxg/nir-dataset-new (accessed Aug. 28, 2025).

[10] H. Agrawal, P. Jain, and A. M. Joshi, "Machine learning models for non-invasive glucose measurement: towards diabetes management in smart healthcare," Health and Technology, vol. 12, no. 5, pp. 955–970, Aug. 2022, doi: https://doi.org/10.1007/s12553-022-00690-7.

[11] M. Naresh, V. Siva Nagaraju, Sreedhar Kollem, J. Kumar, and Samineni Peddakrishna, "Non-invasive Glucose Prediction and Classification Using NIR Technology with Machine Learning," Heliyon, vol. 10, no. 7, pp. e28720–e28720, Apr. 2024, doi: https://doi.org/10.1016/j.heliyon.2024.e28720.

[12] H. A. Muhammed, D. A.-S. Saleh, A. S. Abbas, and A. W. W. Al-Deen, "Designing a Device that Uses Sensor to Measure Blood Sugar Levels," *American Journal of Bioscience and Clinical Integrity*, vol. 2, no. 4, pp. 8–?, Apr. 2025.

[13] A. Nawaz *et al.*, "Review: Non-Invasive Continuous Blood Glucose Measurement Techniques," *Bioinformatics and Diabetes*, vol. 3, pp. 1–?, Jun. 2016, doi: 10.14302/issn.2374-9431.jbd-15-647.

[14] P. Jain, R. Maddila, and A. M. Joshi, "A precise non-invasive blood glucose measurement system using NIR spectroscopy and Huber's regression model," *Opt. Quantum Electron.*, vol. 51, no. 51, Feb. 2019, doi: 10.1007/s11082-019-1751-8.

[15] B. Prastowo, R. P. Jenie, Irzaman, and H. Alatas, "Infra Red – Light Emitting Diode and Photodiode Pair in Measuring Blood Glucose Level Based on Transmittance Method," in *Proc. Int. Conf. Science, Technology, and Environment (ICSTE)*, Aug. 2019, doi: 10.2139/ssrn.3487389

[16] M. S. Hossain *et al.*, "Estimation of Blood Glucose from PPG Signal Using Convolutional Neural Network," in *2019 IEEE Int. Conf. Biomedical Engineering, Computer and Information Technology for Health (BECITHCON)*, Dhaka, Bangladesh, Nov. 2019, pp. 53–58, doi: 10.1109/BECITHCON.2019. something (IEEE Xplore link).

[17] M. Valero *et al.*, "Development of a Noninvasive Blood Glucose Monitoring System Prototype: Pilot Study," *JMIR Formative Research*, vol. 6, no. 8, e38664, Aug. 2022, doi: 10.2196/38664.

[18] W. H. M. Saad *et al.*, "Analysis on Continuous Wearable Device for Blood Glucose Detection Using GSR Sensor," *Int. J. Nanoelectronics and Materials* (in press), 202?.

[19] K. J. Jegadish, P. Kaythry, S. Santhosh, and M. Sheeba, "IoT Based Non-Invasive Blood Glucose Measurement Using Galvanic Skin Response Sensor," *Journal of Next Generation Technology*, vol. 3, no. 1, pp. 23–30, Jul. 2023.

[20] A. E. Hajjaj, F. S. Jaffar, R. W. Alakool, and E. A. Mattar, "Non-Invasive Blood Glucose Level Monitoring for Enhancement using ML," Senior Design Project Report, Dept. Electrical and Electronics Engineering, Univ. of Bahrain, Apr. 2022.

[21] E. Tomé, S. Romanski, N. Freundler, and V. Thavapalan, "A wearable and Non-Invasive Glucose Monitoring Device based on NIR Absorption," Technical Report, Group 18, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, Dec. 2017.

Appendix

LOGBOOK

Student Details	NAME & ID	EMAIL ADDRESS	PHONE
Member 1	NAIM RASHID - 21310009	naim.rashid@g.bracu.ac.bd	01409808215
Member 2	RIFAH TASPIA MITEE - 22121027	rifah.taspia.mitee@g.bracu.ac.bd	01401786188
Member 3	MAYEESHA ISLAM - 22121061	mayeesha.islam@g.bracu.ac.bd	016222819425
Member 4	TASHRIK IBRAHIM - 22121035	tashrik.ibrahim@g.bracu.ac.bd	01971152251
Member 5	ABU ANAS ANTAR - 22121112	abu.anas.antar@g.bracu.ac.bd	01845689121
ATC Details:			
ATC 11			
Chair	Dr. Nahid Akhter Jahan	nahid.jahan@bracu.ac.bd	
Member 1	Dr. Farjana Akter Jhuma	farjana.jhuma@bracu.ac.bd	
Member 2	Subhan Zawad Bihan	subhan.bihan@bracu.ac.bd	

FYDP-P

Sl.	Date/Time/Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
1.	26.06.2025 11:00 AM On Campus (EEE DEPT.)	1.Naim 2.Mitee 3.Mayeesha 4. Tashrik 5. Antar	1. Brief introduction with ATC Chair 2. Instructed to Come Up with 3 project ideas.	All members	● Deliver project ideas by next week. ● ATC Chair will provide sample FYDP-P Report
2.	07.07.2025 11:00 AM On Campus (EEE DEPT.)	1.Naim 2.Mitee 3.Mayeesha 4. Tashrik 5. Antar	1. Proposed multiple project topics to the ATC committee. 2. Getting feedback on our topic.	All members	● Two possible topics were selected by the ATC panel. 1. Was instructed to find novelty in each topic and report back within the following week.
3.	14.07.2025 11:00 AM On Campus (EEE DEPT.)	1.Naim 2.Mitee 3.Mayeesha 4. Tashrik 5. Antar	1. Extensive meeting with ATC panel regarding selected topic. 2. Gained insight on data acquisition and multiple design approaches.	All members	5. Look into the feasibility of the project. 6. Do literature review and identify research gaps.
4.	25.07.2025 11:00 AM Online (Email)	1.Naim 2.Mitee 3.Mayeesha 4. Tashrik 5. Antar	1. Informed ATC about potential topic clash with two other FYDP groups.	All members	● Was instructed to carry on if the design approaches of the different groups were novel.
5.	06.08.25 11:00 AM On Campus (EEE DEPT.)	1.Naim 2.Mitee 3.Mayeesha 4. Tashrik 5. Antar	1. MIDTERM Progress Representation 2. Received Feedback for making necessary corrections on Concept Note	All members	1. Make necessary corrections on requirements, specifications and constraints.
6.	07.08.25 11:00 AM On Campus (EEE DEPT.)	1.Naim 2.Mitee 3.Mayeesha 4. Tashrik 5. Antar	1. Received more instructions to fine tune the details of the concept note.	All members	● Fine tune the concept note report and submit within that night.

7.	20.08.25 2:00 PM On Campus (EEE DEPT.)	1.Naim 2.Mitee 3.Mayeesha 4. Tashrik 5. Antar	1. Instructed to give a mock presentation on the final project report. 2. Prepare a strong argument regarding the novelty of the project.	All members	1. Prepare a first draft of the final report before the mock presentation
8.	21.08.25 11:00 AM On Campus (EEE DEPT.)	1.Naim 2.Mitee 3.Mayeesha 4. Tashrik 5. Antar	1. Selected the date for mock final presentation	All members	2. Prepare a first draft of the final report before the mock presentation
9.	30.08.25 8:00 PM Online	1.Naim 2.Mitee 3.Mayeesha 4. Tashrik 5. Antar	1. Gave Mock Presentation 2. Received Feedback	All members	● Fix multiple design approach and overall flow of the presentation
10.	31.08.25 2:00 PM On Campus (EEE DEPT.)	1.Naim 2.Mitee 3.Mayeesha 4. Tashrik 5. Antar	1. Received Approval on multiple design approach	All members	● Received instruction to prepare adequately for final presentation.
11.	9.9.25 2:00 PM On Campus (EEE DEPT.)	1.Naim 2.Mitee 3.Mayeesha 4. Tashrik 5. Antar	1. Showed first draft of final project proposal	All members	● Received feedback for making final changes

FYDP-D

Date/Time/Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
23.10.205 11:00 AM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Determining the possible simulations that could be done throughout the D phase.	All Members	1. Work on finding relevant designs on the project. 2. Read research papers to get ideas.
28.10.205 12:00 PM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Discussion on the challenges we are facing during the simulation stage.	All Members	
5.11.25 11:30 AM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Identifying the simulations that will be the most essential for this semester.	All Members	
12.11.25 11:00 AM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Identified the 2 sensors we picked for the project from a possible choice of 3 sensors.	All Members	
18.11.25 12:30 PM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Showed the various analog circuits we designed for the simulation and gained feedback.	All Members	
03.12.25 12:30 PM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Official Progress Presentation on the 5 simulations performed throughout the semester.	All Members	Received valuable insights regarding diagrams, tables, and graphs.
07.12.25 12:30 PM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Initial meeting on poster design and design report submission.	All Members	

11.12.25 12:30 PM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Showed initial drafts of the poster design and design report.	All Members	Instructed to make the necessary changes.
14.12.25 2:00 PM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Showed final version of poster design and design report.	All Members	

FYDP-C

Date/Time/Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
01.02.26 10:00 AM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Hardware phase kickoff. Finalised component list: MAX30105 pulse oximeter/PPG sensor, GSR sensor module, Arduino Uno for data acquisition and signal processing, and ESP32 for cloud connectivity. Divided responsibilities among team members.	All Members	Confirm component availability and begin procurement.
06.02.26 11:00 AM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Reviewed MAX30105 and GSR sensor datasheets in detail. Discussed I2C interfacing of MAX30105 with Arduino, and analog voltage divider design for the GSR electrodes. Initiated component orders.	All Members	
11.02.26 11:30 AM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Components received. Set up an Arduino development environment with SparkFun MAX30105 library. Verified basic I2C communication with the MAX30105 sensor and confirmed red/IR LED operation.	All Members	
01.03.26 12:00 PM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Began fieldwork for custom dataset collection. Recruited volunteer subjects for blood glucose and sensor readings. Established a standard measurement protocol: 2-minute rest, 30-second PPG acquisition, simultaneous GSR reading.	All Members	Ensure ethical consent forms are completed for all subjects.
08.03.26 11:30 AM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Fieldwork session 2. Collected readings from additional subjects at varying post-meal intervals. Logged MAX30105 Red/IR raw values and GSR voltage alongside reference glucometer readings into the custom dataset spreadsheet.	All Members	
15.03.26 12:00 PM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Fieldwork session 3 completed, reaching target sample size. Reviewed dataset for missing entries and outliers. Exported CSV for ML team. Initiated ESP32 integration: tested Wi-Fi connection and MQTT-based cloud data publishing.	All Members	Validate ESP32-to-cloud pipeline before next meeting.
22.03.26 11:00 AM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	ESP32 cloud upload pipeline validated. Arduino streams processed sensor data to ESP32 via UART; ESP32 publishes JSON packets to cloud endpoints. Demonstrated live sensor readings appearing on the project website dashboard.	All Members	Improve latency; reduce UART baud rate overhead.
29.03.26 12:30 PM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Website development progress review. Showed patient-facing data entry page, real-time glucose prediction display panel, and historical reading charts. Discussed backend API integration with the ML model inference endpoint.	All Members	Finalise UI before end of April submission.

Date/Time/Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
05.04.26 11:30 AM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Began 3D enclosure design in CAD. Modelled custom holder for the NIR/PPG (MAX30105) sensor module to ensure consistent finger placement and controlled ambient light shielding. Printed first prototype on FDM printer.	All Members	Check sensor fit; tolerances may need adjustment.
12.04.26 12:00 PM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Revised NIR sensor holder after fit test — increased bore diameter by 0.3 mm. Printed second iteration and confirmed snug fit. Designed overall device enclosure box accommodating Arduino, ESP32, battery, and breadboard in a compact form factor.	All Members	
19.04.26 11:00 AM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Printed and assembled final device enclosure. All Breadboard components, Arduino, and ESP32 mounted inside. Performed full end-to-end test: subject places finger in NIR holder, GSR electrodes attached, prediction displayed on website within seconds.	All Members	Received feedback on cable management and labelling.
25.04.26 12:30 PM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Presented a complete hardware prototype to ATC. Demonstrated live glucose prediction using the custom dataset-trained ML model, ESP32 cloud link, and website dashboard. Reviewed website features and documented system architecture.	All Members	Instructed to finalise hardware report and poster.
30.04.26 2:00 PM EEE Dept.	1. Antar 2. Mitee 3. Mayeesha 4. Naim 5. Tashrik	Submitted final hardware phase report, poster, and source code repository. All hardware deliverables (3D-printed enclosure, Arduino/ESP32 firmware, website, and custom dataset) handed over for evaluation.	All Members	

Related Code

1. **Arduino Code**

```
#include <Wire.h>

#define GSR_PIN        A0
#define NUM_SAMPLES    50
#define WARMUP_SAMPLES 15
#define SAMPLE_DELAY   20

#define MAX_ADDR  0x57
#define FIFODATA  0x07
#define MODECONF  0x09
#define FIFOCONF  0x08
#define PARTCONF  0x0A
#define LED1      0x0C
#define LED2      0x0D
#define MULTILED  0x11
#define FIFO_WR   0x04
#define FIFO_RD   0x06

float redRaw[NUM_SAMPLES];
float irRaw[NUM_SAMPLES];
float gsrRaw[NUM_SAMPLES];

void writeReg(uint8_t reg, uint8_t val) {
  Wire.beginTransmission(MAX_ADDR);
  Wire.write(reg); Wire.write(val); Wire.endTransmission();
}
uint8_t readReg(uint8_t reg) {
```

```
  Wire.beginTransmission(MAX_ADDR);
  Wire.write(reg); Wire.endTransmission(false);
  Wire.requestFrom((uint8_t)MAX_ADDR, (uint8_t)1);
  return Wire.read();
}
bool readFIFO(long &red, long &ir) {
  if (readReg(FIFO_WR) == readReg(FIFO_RD)) return false;
  Wire.beginTransmission(MAX_ADDR);
  Wire.write(FIFODATA); Wire.endTransmission(false);
  Wire.requestFrom((uint8_t)MAX_ADDR, (uint8_t)6);
  red =
((long)(Wire.read()&0x03)<<16)|((long)Wire.read()<<8)|Wire.read();
  ir  =
((long)(Wire.read()&0x03)<<16)|((long)Wire.read()<<8)|Wire.read();
  return true;
}
void clearFIFO() {
  writeReg(FIFO_WR, 0x00);
  writeReg(FIFO_RD, 0x00);
  writeReg(0x05,    0x00);
}
void initSensor() {
  writeReg(MODECONF, 0x40); delay(100);
  writeReg(FIFOCONF, 0x50);
  writeReg(MODECONF, 0x03);
  writeReg(PARTCONF, 0x27);
  writeReg(LED1,     0x1F);
  writeReg(LED2,     0x1F);
  writeReg(MULTILED, 0x21);
```

```
  clearFIFO(); delay(100);
}

float calcMean(float* d, int n) {
  float s = 0;
  for (int i = 0; i < n; i++) s += d[i];
  return s / n;
}
float calcStd(float* d, int n, float mean) {
  float s = 0;
  for (int i = 0; i < n; i++) s += (d[i]-mean)*(d[i]-mean);
  return sqrt(s / n);
}
void getStats(float* data, float &outMean, float &outSD) {
  float m  = calcMean(data, NUM_SAMPLES);
  float s  = calcStd(data, NUM_SAMPLES, m);
  float lo = m - 2.0*s;
  float hi = m + 2.0*s;
  float filtered[NUM_SAMPLES];
  int fc = 0;
  for (int i = 0; i < NUM_SAMPLES; i++)
    if (data[i] >= lo && data[i] <= hi) filtered[fc++] = data[i];
  outMean = calcMean(filtered, fc);
  outSD   = calcStd(filtered, fc, outMean);
}

bool collectNIR() {
  clearFIFO(); delay(100);
  int count = 0;
```

```
  while (count < WARMUP_SAMPLES) {
    long r, ir;
    if (readFIFO(r, ir)) { count++; delay(SAMPLE_DELAY); }
  }
  count = 0;
  while (count < NUM_SAMPLES) {
    long r, ir;
    if (readFIFO(r, ir)) {
      redRaw[count] = (float)r;
      irRaw[count]  = (float)ir;
      count++; delay(SAMPLE_DELAY);
    }
  }
  return true;
}
void collectGSR() {
  for (int i = 0; i < WARMUP_SAMPLES; i++) {
    analogRead(GSR_PIN); delay(SAMPLE_DELAY);
  }
  for (int i = 0; i < NUM_SAMPLES; i++) {
    gsrRaw[i] = (float)analogRead(GSR_PIN);
    delay(SAMPLE_DELAY);
  }
}

void runMeasurement() {
  //Serial.println(F("Collecting NIR..."));
  collectNIR();
  //Serial.println(F("Collecting GSR..."));
```

```
  collectGSR();

  float redMean, redSD, irMean, irSD, gsrMean, gsrSD;
  getStats(redRaw, redMean, redSD);
  getStats(irRaw,  irMean,  irSD);
  getStats(gsrRaw, gsrMean, gsrSD);

  gsrSD = gsrSD * 0.1;

  Serial.print(redMean, 2); Serial.print(',');
  Serial.print(redSD,   2); Serial.print(',');
  Serial.print(irMean,  2); Serial.print(',');
  Serial.print(irSD,    2); Serial.print(',');
  Serial.print(gsrMean, 2); Serial.print(',');
  Serial.println(gsrSD, 2);
}

void setup() {
  Serial.begin(115200);
  Wire.begin();
  initSensor();
  Serial.println(F("ARDUINO_READY"));
}

void loop() {
  if (Serial.available()) {
    String cmd = Serial.readStringUntil('\n');
    cmd.trim();
    if (cmd == "START") {
```

```
      runMeasurement();
    }
  }
}
```

2. ESP32 Code

```
#include <U8g2lib.h>
#include <Keypad.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <SPI.h>

const char* WIFI_SSID     = "OP7T";
const char* WIFI_PASSWORD = "#naim123#";
const char* SERVER_IP     = "10.159.78.24";
const int   SERVER_PORT   = 5000;

#define BTN_MEASURE 15

#define OLED_CLK   18
#define OLED_MOSI  23
#define OLED_CS    19
#define OLED_DC     0
#define OLED_RES    2

#define ARD_RX 16
#define ARD_TX 17
```

```
U8G2_SSD1309_128X64_NONAME0_1_4W_SW_SPI u8g2(
  U8G2_R0,
  OLED_CLK, OLED_MOSI, OLED_CS, OLED_DC, OLED_RES
);

const byte ROWS = 4, COLS = 3;
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
byte rowPins[ROWS] = {13, 12, 14, 27};
byte colPins[COLS]  = {26, 25, 33};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS,
COLS);

int  userAge, userGender, userMealTime;
int  userSkinTone, userDiabetes, userDiabHistory;
long  redMean, irMean, gsrMean;
float redSD, irSD, gsrSD;

enum State { STATE_INPUT, STATE_WAIT_MEASURE, STATE_MEASURING,
STATE_SENDING, STATE_RESULT };
State currentState = STATE_INPUT;

float lastGlucose = 0;

void oledShow(const char* l1, const char* l2=nullptr,
```

```
              const char* l3=nullptr, const char* l4=nullptr) {
  u8g2.firstPage();
  do {
    u8g2.setFont(u8g2_font_ncenB08_tr);
    if (l1) u8g2.drawStr(0, 14, l1);
    if (l2) u8g2.drawStr(0, 28, l2);
    if (l3) u8g2.drawStr(0, 42, l3);
    if (l4) u8g2.drawStr(0, 56, l4);
  } while (u8g2.nextPage());
}

void showGlucoseResult(float glucose) {
  char l1[22], l2[22];
  snprintf(l1, sizeof(l1), "%.1f mmol/L", glucose);
  if      (glucose < 3.9)  strcpy(l2, "LOW-Hypoglycemia");
  else if (glucose <= 5.6) strcpy(l2, "Normal (Fasting)");
  else if (glucose <= 7.8) strcpy(l2, "Pre-diabetic");
  else                     strcpy(l2, "HIGH-Check again");
  u8g2.firstPage();
  do {
    u8g2.setFont(u8g2_font_5x7_tr);
    u8g2.drawStr(0,  8, l1);
    u8g2.drawStr(0, 20, l2);
    u8g2.drawStr(0, 36, "# = Remeasure");
    u8g2.drawStr(0, 48, "* = Send to ML");
    u8g2.drawStr(0, 60, "BTN = Full Reset");
  } while (u8g2.nextPage());
}
```

```
void connectWiFi() {
  oledShow("Connecting WiFi", WIFI_SSID, "Please wait...", "");
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  int attempts = 0;
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); attempts++;
    if (attempts > 40) {
      oledShow("WiFi FAILED!", "Check name &", "password",
"Restarting...");
      delay(3000); ESP.restart();
    }
  }
  char ipStr[22];
  snprintf(ipStr, sizeof(ipStr), "%s",
WiFi.localIP().toString().c_str());
  Serial.print("WiFi connected. IP: "); Serial.println(ipStr);
  oledShow("WiFi Connected!", ipStr, "", "");
  delay(2000);
}

bool collectFromArduino() {
  while (Serial2.available()) Serial2.read();
  Serial2.println("START");
  Serial.println("Sent START to Arduino");
  oledShow("Measuring...", "Hold finger on", "NIR sensor!", "~5
seconds");

  unsigned long t0 = millis();
  String line = "";
```

```
  while (millis() - t0 < 12000) {
    if (Serial2.available()) {
      char c = Serial2.read();
      if (c == '\n') break;
      if (c != '\r') line += c;
    }
  }

  Serial.print("Arduino response: "); Serial.println(line);

  if (line.length() == 0) {
    oledShow("Arduino timeout!", "No response", "Check wiring",
"TX/RX/GND");
    delay(2000); return false;
  }

  int commaCount = 0;
  for (char c : line) if (c == ',') commaCount++;

  if (commaCount != 5) {
    unsigned long t1 = millis();
    line = "";
    while (millis() - t1 < 8000) {
      if (Serial2.available()) {
        char c = Serial2.read();
        if (c == '\n') {
          commaCount = 0;
          for (char ch : line) if (ch == ',') commaCount++;
          if (commaCount == 5) break;
```

```
        line = "";
      } else if (c != '\r') {
        line += c;
      }
    }
  }
  }

  if (commaCount != 5) {
    oledShow("Parse error!", "Bad data from", "Arduino", "Try
again");
    delay(2000); return false;
  }

  float vals[6];
  int idx = 0;
  char buf[line.length() + 1];
  line.toCharArray(buf, sizeof(buf));
  char* tok = strtok(buf, ",");
  while (tok && idx < 6) {
    vals[idx++] = atof(tok);
    tok = strtok(nullptr, ",");
  }
  if (idx < 6) {
    oledShow("Parse error!", "Incomplete data", "", "Try again");
    delay(2000); return false;
  }

  redMean = (long)vals[0];
```

```
  redSD   = vals[1];
  irMean  = (long)vals[2];
  irSD    = vals[3];
  gsrMean = (long)vals[4];
  gsrSD   = vals[5];

  Serial.printf("redM=%ld redSD=%.2f irM=%ld irSD=%.2f gsrM=%ld
gsrSD=%.2f\n",
    redMean, redSD, irMean, irSD, gsrMean, gsrSD);
  return true;
}

float sendToMLServer() {
  if (WiFi.status() != WL_CONNECTED) connectWiFi();
  HTTPClient http;
  char url[60];
  snprintf(url, sizeof(url), "http://%s:%d/predict", SERVER_IP,
SERVER_PORT);
  http.begin(url);
  http.addHeader("Content-Type", "application/json");
  http.setTimeout(15000);

  char rs[10], is_[10], gs[10];
  dtostrf(redSD, 6, 2, rs);
  dtostrf(irSD,  6, 2, is_);
  dtostrf(gsrSD, 6, 2, gs);

  char payload[300];
  snprintf(payload, sizeof(payload),
```

```
    "{\"age\":%d,\"gender\":%d,\"meal\":%d,\"skin\":%d,"
    "\"diab\":%d,\"diabH\":%d,"
    "\"redM\":%ld,\"redSD\":%s,"
    "\"irM\":%ld,\"irSD\":%s,"
    "\"gsrM\":%ld,\"gsrSD\":%s}",
    userAge, userGender, userMealTime, userSkinTone,
    userDiabetes, userDiabHistory,
    redMean, rs, irMean, is_, gsrMean, gs
  );

  Serial.println("Sending:"); Serial.println(payload);
  oledShow("Sending to", "ML server...", "Please wait", "");
  int httpCode = http.POST(payload);
  Serial.print("HTTP: "); Serial.println(httpCode);

  if (httpCode == 200) {
    String response = http.getString();
    Serial.println("Response: " + response);
    http.end();
    StaticJsonDocument<64> doc;
    if (!deserializeJson(doc, response) &&
doc.containsKey("glucose"))
      return doc["glucose"].as<float>();
    return -2;
  }
  http.end();
  return -1;
}
```

```
int getNumberInput(const char* question, const char* hint) {
  char inputBuf[6] = {0}; int len = 0;
  while (true) {
    char display[22]; strcpy(display, "> ");
    strcat(display, len > 0 ? inputBuf : "_");
    oledShow(question, hint, display, "#=OK  *=Back");
    char key = keypad.getKey();
    if (key) {
      if      (key >= '0' && key <= '9') { if (len < 5) {
inputBuf[len++] = key; inputBuf[len] = '\0'; } }
      else if (key == '*') { if (len > 0) inputBuf[--len] = '\0'; }
      else if (key == '#') { if (len > 0) return atoi(inputBuf); }
    }
    delay(20);
  }
}


bool collectUserInputs() {
  int val;
  while(true){val=getNumberInput("Age (years):","");
if(val>=1&&val<=120){userAge=val;break;}       oledShow("Invalid
age!","Enter 1-120","","");     delay(1500);}
  while(true){val=getNumberInput("Gender:","1=Male  0=Female");
if(val==0||val==1){userGender=val;break;}
oledShow("Invalid!","Enter 0 or 1","","");       delay(1500);}
  while(true){val=getNumberInput("Hrs since meal:","e.g. 2");
if(val>=0&&val<=24){userMealTime=val;break;}
oledShow("Invalid!","Enter 0-24 hrs","","");     delay(1500);}
  while(true){val=getNumberInput("Skin tone:","1=Fair 2=Med
3=Drk");if(val>=1&&val<=3){userSkinTone=val;break;}
oledShow("Invalid!","Enter 1,2 or 3","","");     delay(1500);}
```

```
  while(true){val=getNumberInput("Diabetes?","1=Yes  0=No");
if(val==0||val==1){userDiabetes=val;break;}
oledShow("Invalid!","Enter 0 or 1","","");       delay(1500);}

  while(true){val=getNumberInput("Diab. history?","1=Yes  0=No");
if(val==0||val==1){userDiabHistory=val;break;}oledShow("Invalid!","E
nter 0 or 1","","");       delay(1500);}

  return true;

}


bool btnPressed() {

  if (digitalRead(BTN_MEASURE) == LOW) {

    delay(50);

    if (digitalRead(BTN_MEASURE) == LOW) {

      while (digitalRead(BTN_MEASURE) == LOW);

      return true;

    }

  }

  return false;

}


void setup() {

  Serial.begin(115200);

  delay(500);

  pinMode(BTN_MEASURE, INPUT_PULLUP);


  u8g2.begin();

  oledShow("Glucose Monitor", "Initializing...", "", "");


  Serial2.begin(115200, SERIAL_8N1, ARD_RX, ARD_TX);

  delay(2000);
```

```
  oledShow("Waiting for", "Arduino...", "", "");
  unsigned long t0 = millis();
  bool ardReady = false;
  while (millis() - t0 < 5000) {
    if (Serial2.available()) {
      String line = Serial2.readStringUntil('\n');
      line.trim();
      if (line == "ARDUINO_READY") { ardReady = true; break; }
    }
  }
  if (!ardReady) {
    oledShow("Arduino?", "No READY msg", "Continuing...", "");
    delay(1500);
  }

  connectWiFi();
  oledShow("Glucose Monitor", "Ready!", "", "");
  delay(1000);
}

void loop() {

  // --- Hard reset from anywhere ---
  if (btnPressed()) {
    currentState = STATE_INPUT;
    oledShow("Resetting...", "", "", "");
    delay(1000);
    return;
```

```
}

// --- STATE: Collect user inputs via keypad ---
if (currentState == STATE_INPUT) {
  collectUserInputs();
  currentState = STATE_WAIT_MEASURE;
  return;
}

// --- STATE: Wait for # to start measurement ---
if (currentState == STATE_WAIT_MEASURE) {
  oledShow("Place hand on", "sensor", "Press # to", "measure");
  char key = keypad.getKey();
  if (key == '#') {
    currentState = STATE_MEASURING;
  }
  return;
}

// --- STATE: Collect data from Arduino ---
if (currentState == STATE_MEASURING) {
  if (collectFromArduino()) {
    currentState = STATE_SENDING;
  } else {
    // measurement failed, go back to wait screen
    currentState = STATE_WAIT_MEASURE;
  }
  return;
}
```

```
  // --- STATE: Send to ML server ---
  if (currentState == STATE_SENDING) {
    float glucose = sendToMLServer();
    if (glucose < 0) {
      oledShow("Server error!", "Check laptop", "Press # retry",
"BTN = Reset");
      char key = keypad.getKey();
      if (key == '#') currentState = STATE_SENDING;
      return;
    }
    lastGlucose = glucose;
    showGlucoseResult(glucose);
    currentState = STATE_RESULT;
    return;
  }

  // --- STATE: Show result, wait for action ---
  if (currentState == STATE_RESULT) {
    char key = keypad.getKey();
    if (key == '#') {
      // Remeasure with same user inputs
      currentState = STATE_WAIT_MEASURE;
    } else if (key == '*') {
      // Re-send existing data to ML (already sent, just show again
or re-send)
      currentState = STATE_SENDING;
    }
    return;
```

```
  }
}
```

3. Server Code

```
from flask import Flask, request, jsonify
import joblib
import pandas as pd
import numpy as np

app = Flask(__name__)

# Load model
model = joblib.load('best_model.pkl')

# Training-time normalization constants (from notebook)
MEAN_RED_MEAN = 80164.88826815642
STD_RED_MEAN  = 14946.940562635956
MEAN_IR_MEAN  = 90623.75418994413
STD_IR_MEAN   = 12154.27788421458
MEAN_GSR_MEAN = 585.804469273743
STD_GSR_MEAN  = 48.0678650042456

feature_cols = [
    'Age', 'Gender', 'Time Since Last Meal (Hr)', 'Skin Tone',
    'Red Mean', 'Red SD', 'IR Mean', 'IR SD', 'Red/IR',
    'GSR Mean', 'GSR SD', 'Diabetes', 'Diabetic History',
    'Red_CV', 'IR_CV', 'GSR_CV', 'Red_Stability', 'IR_Stability',
    'RedIR_Age', 'RedIR_Diabetes', 'GSR_Diabetes', 'Meal_RedIR',
    'Age_Diabetes', 'Log_RedMean', 'Log_IRMean', 'Log_GSRMean',
    'Red_IR_Diff', 'Red_IR_SD_Diff', 'Diabetes_Risk',
```

```
    'Red_Normalized', 'IR_Normalized', 'GSR_Normalized',
    'Age_Group', 'Fasting', 'Recent_Meal'
]

@app.route('/predict', methods=['POST'])
def predict():
    data = request.json

    raw = pd.DataFrame([{
        'Age':                        data['age'],
        'Gender':                     data['gender'],
        'Time Since Last Meal (Hr)': data['meal'],
        'Skin Tone':                  data['skin'],
        'Red Mean':                   data['redM'],
        'Red SD':                     data['redSD'],
        'IR Mean':                    data['irM'],
        'IR SD':                      data['irSD'],
        'GSR Mean':                   data['gsrM'],
        'GSR SD':                     data['gsrSD'],
        'Diabetes':                   data['diab'],
        'Diabetic History':           data['diabH'],
    }])

    df = raw.copy()
    df['Red/IR']         = df['Red Mean'] / (df['IR Mean'] + 1e-6)
    df['Red_CV']         = df['Red SD']   / (df['Red Mean'] + 1e-6)
    df['IR_CV']          = df['IR SD']    / (df['IR Mean'] + 1e-6)
    df['GSR_CV']         = df['GSR SD']   / (df['GSR Mean'] + 1e-6)
    df['Red_Stability']  = df['Red Mean'] / (df['Red SD'] + 1e-6)
```

```
    df['IR_Stability']   = df['IR Mean']  / (df['IR SD']  + 1e-6)
    df['RedIR_Age']      = df['Red/IR']   * df['Age']
    df['RedIR_Diabetes'] = df['Red/IR']   * df['Diabetes']
    df['GSR_Diabetes']   = df['GSR Mean'] * df['Diabetes']
    df['Meal_RedIR']     = df['Time Since Last Meal (Hr)'] *
df['Red/IR']
    df['Age_Diabetes']   = df['Age']      * df['Diabetes']
    df['Log_RedMean']    = np.log1p(df['Red Mean'])
    df['Log_IRMean']     = np.log1p(df['IR Mean'])
    df['Log_GSRMean']    = np.log1p(df['GSR Mean'])
    df['Red_IR_Diff']    = df['Red Mean'] - df['IR Mean']
    df['Red_IR_SD_Diff'] = df['Red SD']   - df['IR SD']
    df['Diabetes_Risk']  = df['Diabetes'] + df['Diabetic History']
    df['Red_Normalized'] = (df['Red Mean'] - MEAN_RED_MEAN) /
STD_RED_MEAN
    df['IR_Normalized']  = (df['IR Mean']  - MEAN_IR_MEAN)  /
STD_IR_MEAN
    df['GSR_Normalized'] = (df['GSR Mean'] - MEAN_GSR_MEAN) /
STD_GSR_MEAN
    df['Age_Group']      = pd.cut(df['Age'], bins=[0, 25, 35, 50,
100],
                                  labels=[0, 1, 2, 3]).astype(float)
    df['Fasting']        = (df['Time Since Last Meal (Hr)'] >=
8).astype(int)
    df['Recent_Meal']    = (df['Time Since Last Meal (Hr)'] <=
2).astype(int)

    X = df[feature_cols]
    pred = model.predict(X)[0]

    # Offset correction
```

```
    #if 7 < pred < 10:
    #    pred += 2
    #elif pred >= 9.5:
    #    pred += 3.5
    if pred >= 9.5:
        pred +=3.5

    print(f"Input: age={data['age']} gender={data['gender']}
meal={data['meal']} skin={data['skin']}")
    print(f"Sensor: redM={data['redM']} redSD={data['redSD']}
irM={data['irM']} irSD={data['irSD']} gsrM={data['gsrM']}
gsrSD={data['gsrSD']}")
    print(f"Predicted glucose: {pred:.2f} mmol/L")

    return jsonify({'glucose': round(float(pred), 2)})

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

4. Machine Learning Model

```
import re
import warnings
import joblib
import numpy as np
import pandas as pd

from sklearn.model_selection import (
    train_test_split,
    KFold,
    cross_validate
```

```
)

from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

from sklearn.preprocessing import (
    OneHotEncoder,
    RobustScaler
)

from sklearn.impute import SimpleImputer

from sklearn.metrics import (
    mean_absolute_error,
    mean_squared_error,
    r2_score
)

from sklearn.ensemble import (
    RandomForestRegressor,
    GradientBoostingRegressor,
    ExtraTreesRegressor,
    StackingRegressor,
    VotingRegressor
)

from sklearn.linear_model import (
    Ridge,
    BayesianRidge,
```

```
    ElasticNet
)

from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor

from xgboost import XGBRegressor
from lightgbm import LGBMRegressor
from catboost import CatBoostRegressor

import optuna

warnings.filterwarnings("ignore")
optuna.logging.set_verbosity(optuna.logging.WARNING)

file_name = "glucose_dataset.xlsx"

if file_name.lower().endswith(".csv"):
    df = pd.read_csv(file_name)
elif file_name.lower().endswith((".xlsx", ".xls")):
    df = pd.read_excel(file_name)
else:
    raise ValueError("Dataset must be in CSV or Excel format.")

df.columns = [
    re.sub(r"\s+", " ", column.strip())
    for column in df.columns
]
```

```
print(
    f"Dataset loaded successfully: "
    f"{df.shape[0]} rows × {df.shape[1]} columns"
)

TARGET = "Blood Glucose"

df["Gender"] = (
    df["Gender"]
    .astype(str)
    .str.strip()
    .str.lower()
    .map({"male": 1, "female": 0})
)

df["Diabetes"] = (
    df["Diabetes"]
    .astype(str)
    .str.strip()
    .str.lower()
    .map({"yes": 1, "no": 0})
)

df["Diabetic History"] = (
    df["Diabetic History"]
    .astype(str)
    .str.strip()
    .str.lower()
    .map({"yes": 1, "no": 0})
```

```
)

df[TARGET] = pd.to_numeric(df[TARGET], errors="coerce")
df["Skin Tone"] = pd.to_numeric(df["Skin Tone"], errors="coerce")

print("Categorical transformation completed.")

df_clean = df.copy()

df_clean = df_clean[df_clean["IR Mean"] >= 50000]

df_clean = df_clean[
    (df_clean["Red/IR"] >= 0.55) &
    (df_clean["Red/IR"] <= 1.20)
]

df_clean = df_clean[df_clean["GSR Mean"] != 0]

df_clean = df_clean[
    ~(
        (df_clean["Blood Glucose"] > 9) &
        (df_clean["Diabetes"] == 0) &
        (df_clean["Diabetic History"] == 0)
    )
]

df_clean = df_clean[df_clean["Blood Glucose"] <= 22]

df_clean = df_clean[df_clean["Red SD"] <= 350]
```

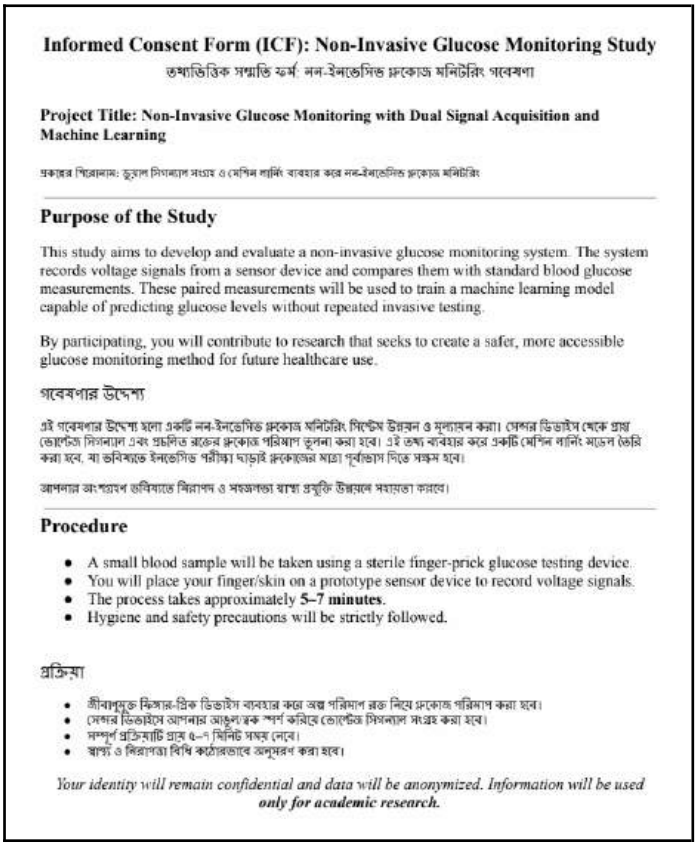
```
print(
    f"Cleaned dataset size: {len(df_clean)} samples "
    f"(removed {len(df) - len(df_clean)} outliers)"
)

df_feat = df_clean.copy()

df_feat["Red_CV"] = (
    df_feat["Red SD"] /
    (df_feat["Red Mean"] + 1e-6)
)

df_feat["IR_CV"] = (
    df_feat["IR SD"] /
    (df_feat["IR Mean"] + 1e-6)
)

df_feat["GSR_CV"] = (
    df_feat["GSR SD"] /
    (df_feat["GSR Mean"] + 1e-6)
)

df_feat["Red_Stability"] = (
    df_feat["Red Mean"] /
    (df_feat["Red SD"] + 1e-6)
)

df_feat["IR_Stability"] = (
```

```
    df_feat["IR Mean"] /
    (df_feat["IR SD"] + 1e-6)
)

df_feat["RedIR_Age"] = df_feat["Red/IR"] * df_feat["Age"]
df_feat["RedIR_Diabetes"] = df_feat["Red/IR"] * df_feat["Diabetes"]
df_feat["GSR_Diabetes"] = df_feat["GSR Mean"] * df_feat["Diabetes"]

df_feat["Meal_RedIR"] = (
    df_feat["Time Since Last Meal (Hr)"] *
    df_feat["Red/IR"]
)

df_feat["Age_Diabetes"] = (
    df_feat["Age"] *
    df_feat["Diabetes"]
)

df_feat["Log_RedMean"] = np.log1p(df_feat["Red Mean"])
df_feat["Log_IRMean"] = np.log1p(df_feat["IR Mean"])
df_feat["Log_GSRMean"] = np.log1p(df_feat["GSR Mean"])

df_feat["Red_IR_Diff"] = (
    df_feat["Red Mean"] -
    df_feat["IR Mean"]
)

df_feat["Red_IR_SD_Diff"] = (
    df_feat["Red SD"] -
```

```
    df_feat["IR SD"]
)

df_feat["Diabetes_Risk"] = (
    df_feat["Diabetes"] +
    df_feat["Diabetic History"]
)

df_feat["Red_Normalized"] = (
    (df_feat["Red Mean"] - df_feat["Red Mean"].mean()) /
    df_feat["Red Mean"].std()
)

df_feat["IR_Normalized"] = (
    (df_feat["IR Mean"] - df_feat["IR Mean"].mean()) /
    df_feat["IR Mean"].std()
)

df_feat["GSR_Normalized"] = (
    (df_feat["GSR Mean"] - df_feat["GSR Mean"].mean()) /
    df_feat["GSR Mean"].std()
)

df_feat["Age_Group"] = pd.cut(
    df_feat["Age"],
    bins=[0, 25, 35, 50, 100],
    labels=[0, 1, 2, 3]
).astype(float)
```

```
df_feat["Fasting"] = (
    df_feat["Time Since Last Meal (Hr)"] >= 8
).astype(int)

df_feat["Recent_Meal"] = (
    df_feat["Time Since Last Meal (Hr)"] <= 2
).astype(int)
```

Informed Consent Form



Informed Consent Form (ICF): Non-Invasive Glucose Monitoring Study

তথ্যভিত্তিক সম্মতি ফর্ম: নন-ইনভেসিভ গ্লুকোজ মনিটরিং গবেষণা

Project Title: Non-Invasive Glucose Monitoring with Dual Signal Acquisition and Machine Learning

প্রকল্পের শিরোনাম: ডুয়াল সিগন্যাল সংগ্রহ ও মেশিন লার্নিং ব্যবহার করে নন-ইনভেসিভ গ্লুকোজ মনিটরিং

Purpose of the Study

This study aims to develop and evaluate a non-invasive glucose monitoring system. The system records voltage signals from a sensor device and compares them with standard blood glucose measurements. These paired measurements will be used to train a machine learning model capable of predicting glucose levels without repeated invasive testing.

By participating, you will contribute to research that seeks to create a safer, more accessible glucose monitoring method for future healthcare use.

গবেষণার উদ্দেশ্য

এই গবেষণার উদ্দেশ্য হলো একটি নন-ইনভেসিভ গ্লুকোজ মনিটরিং সিস্টেম উন্নয়ন ও মূল্যায়ন করা। সেন্সর ডিভাইস থেকে প্রাপ্ত ভোল্টেজ সিগন্যাল এবং প্রচলিত রক্তের গ্লুকোজ পরিমাপ তুলনা করা হবে। এই তথ্য ব্যবহার করে একটি মেশিন লার্নিং মডেল তৈরি করা হবে, যা ভবিষ্যতে ইনভেসিভ পরীক্ষা ছাড়াই গ্লুকোজের মাত্রা পূর্বাভাস দিতে সক্ষম হবে।

আপনার অংশগ্রহণ ভবিষ্যতে শিরাপাদ ও সহজলভ্য স্বাস্থ্য প্রযুক্তি উন্নয়নে সহায়তা করবে।

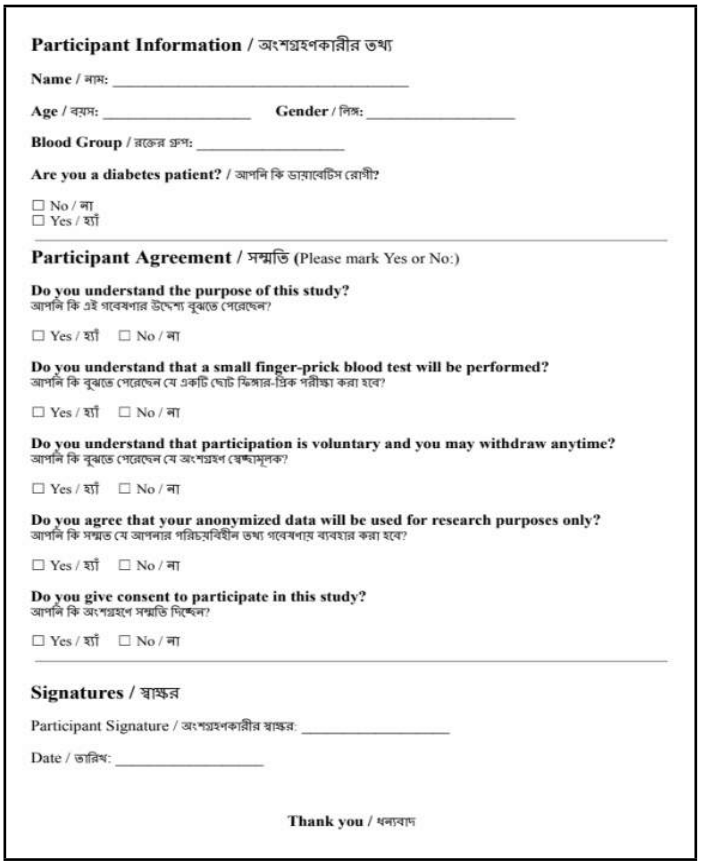
Procedure

- A small blood sample will be taken using a sterile finger-prick glucose testing device.
- You will place your finger/skin on a prototype sensor device to record voltage signals.
- The process takes approximately **5–7 minutes**.
- Hygiene and safety precautions will be strictly followed.

প্রক্রিয়া

- জীবাণুমুক্ত ফিঙ্গার-প্রিক ডিভাইস ব্যবহার করে অল্প পরিমাণ রক্ত নিয়ে গ্লুকোজ পরিমাপ করা হবে।
- সেন্সর ডিভাইসে আপনার আঙুলস্পর্শ করিয়ে ভোল্টেজ সিগন্যাল সংগ্রহ করা হবে।
- সম্পূর্ণ প্রক্রিয়াটি প্রায় ৫–৭ মিনিট সময় নেবে।
- স্বাস্থ্য ও নিরাপত্তা বিধি কঠোরভাবে অনুসরণ করা হবে।

*Your identity will remain confidential and data will be anonymized. Information will be used **only for academic research.***



Participant Information / অংশগ্রহণকারীর তথ্য

Name / নাম: _____

Age / বয়স: _____ **Gender** / লিঙ্গ: _____

Blood Group / রক্তের গ্রুপ: _____

Are you a diabetes patient? / আপনি কি ডায়াবেটিস রোগী?

☐ No / না
☐ Yes / হ্যাঁ

Participant Agreement / সম্মতি (Please mark Yes or No:)

Do you understand the purpose of this study?
আপনি কি এই গবেষণার উদ্দেশ্য বুঝতে পেরেছেন?

☐ Yes / হ্যাঁ ☐ No / না

Do you understand that a small finger-prick blood test will be performed?
আপনি কি বুঝতে পেরেছেন যে একটি ছোট ফিঙ্গার-প্রিক পরীক্ষা করা হবে?

☐ Yes / হ্যাঁ ☐ No / না

Do you understand that participation is voluntary and you may withdraw anytime?
আপনি কি বুঝতে পেরেছেন যে অংশগ্রহণ স্বেচ্ছামূলক?

☐ Yes / হ্যাঁ ☐ No / না

Do you agree that your anonymized data will be used for research purposes only?
আপনি কি সম্মত যে আপনার পরিচয়বিহীন তথ্য গবেষণায় ব্যবহার করা হবে?

☐ Yes / হ্যাঁ ☐ No / না

Do you give consent to participate in this study?
আপনি কি অংশগ্রহণে সম্মতি দিচ্ছেন?

☐ Yes / হ্যাঁ ☐ No / না

Signatures / স্বাক্ষর

Participant Signature / অংশগ্রহণকারীর স্বাক্ষর: _____

Date / তারিখ: _____

Thank you / ধন্যবাদ

Assessment Guideline for Faculty

[The following assessment guideline is for faculty ONLY. **This portion is not applicable for students**.]

Assessment Tools and CO Assessment Guideline

	Distribution of assessment points among various COs assessed in different semesters														
PO	l	c	f	g	c	b	d	c	e	l	k	k	h	i	j
CO	CO1	CO2	CO3	CO4	CO5	CO6	CO7	CO8	CO9	CO10	CO11	CO12	CO13	CO14	CO15
EEE 400C/ ECE 402C (Out of 100)							30	24	6	4	4	6	7	7	12
Project Final Report/ Project Progress Report							x	x	x	x	x	x	x		x
Demonstration of working prototype							x								x
Progress Presentation/ Final Presentation								x			x				
Peer-evaluation*													x	x	
Instructor's Assessment*													x	x	
Demonstration at FYDP Showcase								x							x

Note: The star (*) marked deliverables/skills will be evaluated at various stages of the project.

Mapping of CO-PO-Taxonomy Domain & Level- Delivery-Assessment Tool

Sl.	CO Description	PO	Bloom's Taxonomy Domain/Level	Assessment Tools
CO7	**Evaluate** the performance of the developed solution with respect to the given specifications, requirements and standards	d	Cognitive/ Evaluate	● Demonstration of working prototype ● Project Progress Report on working prototype
CO8	**Complete** the final design and development of the solution with necessary adjustment based on performance evaluation	c	Cognitive/ Create	● Project Final Report ● Final Presentation ● Demonstration at FYDP Showcase
CO9	**Use** modern engineering and IT tools to design, develop and validate the solution	e	Cognitive/ Understand, Psychomotor/ Precision	● Project Final Report
CO10	**Conduct** independent research, literature survey and learning of new technologies and concepts as appropriate to design, develop and validate the solution	l	Cognitive/ Apply	● Project Final Report
CO11**	**Demonstrate** project management skill in various stages of developing the solution of engineering design project	k	Cognitive/ Apply Affective/ Valuing	● Project Final Report ● Project Progress presentation at various stages
CO12	**Perform** cost-benefit and economic analysis of the solution	k	Cognitive/ Apply	● Project Final Report
CO13	**Apply** ethical considerations and professional responsibilities in designing the solution and	h	Cognitive/ Apply Affective/ Valuing	● Peer-evaluation,

	throughout the project development phases			• Instructor's Assessment • Final Report
CO14**	**Perform** effectively as an individual and as a team member for successfully completion of the project	i	Affective/ Characterization	• Peer-evaluation • Instructor's Assessment
CO15**	**Communicate** effectively through writings, journals, technical reports, deliverables, presentations and verbal communication as appropriate at various stages of project development	j	Cognitive/ Understand Psychomotor/ Precision Affective/ Valuing	• Project Final Report • Progress Presentations, • Final Presentation • Demonstration at FYDP Showcase

Note: The double star (**) marked CO will be assessed at various stages of the project through indirect deliverables.