DEVELOPMENT OF A STROKE PREDICTION SYSTEM BASED ON IOT AND MACHINE LEARNING

SABEKUN NAHAR REFAT

M. Engineering THESIS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MILITERY INSTITUTE OF SCIENCE AND TECHNOLOGY
DHAKA, BANGLADESH

MARCH 2024

DEVELOPMENT OF A STROKE PREDICTION SYSTEM BASED ON IOT AND MACHINE LEARNING

SABEKUN NAHAR REFAT (SN 0418140021)

A Project Submitted in Partial Fulfillment of the Degree of Master of Engineering in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MILITARY INSTITUTE OF SCIENCE AND TECHNOLOGY DHAKA, BANGLADESH

MARCH 2024

DEVELOPMENT OF A STROKE PREDICTION SYSTEM BASED ON IOT AND MACHINE LEARNING

M. ENGINEERING PROJECT
BY
SABEKUN MAHAR REFAT (SN 0418140021)

Approved as to style and content by the board of examination on 28 March 2024

Dr. A.K.M. Muzahidul Islam Professor Department of CSE, UIU, Dhaka.	Chairman (Supervisor) Board of examination
Dr. Hosney Jahan Assistant Professor Department of Computer Science and Engineering, Military Institute of Science and Technology.	Member (Internal) Board of Examination
Dr. Muhammad Golam Kibria Professor Department of Computer Science and Engineering, University of Liberal Arts Bangladesh.	Member (External) Board of Examination
Brig Gen Mohammad Sajjad Hossain Head of the Department Department of CSE, MIST, Dhaka.	Head of the department Board of Examination

Department of Computer Science and Engineering, MIST, Dhaka.

DEVELOPMENT OF A STROKE PREDICTION SYSTEM BASED ON IOT AND MACHINE LEARNING

DECLARATION

I hereby declare that this project is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the project. The project (fully or partially) has not been submitted for any degree or diploma in any university or institute previously.

_____

Sabekun Nahar Refat

Department of Computer Science and Engineering, MIST, Dhaka.

ABSTRACT

DEVELOPMENT OF A STROKE PREDICTION SYSTEM BASED ON IOT AND MACHINE LEARNING

Stroke is one of the leading causes of disability in many Asian countries, with low and middle-income countries bearing a higher burden of mortality. Worldwide Cerebrovascular accidents (stroke) are the second leading cause of death and the third leading cause of disability, where in Bangladesh it is the third leading cause of death. Effective prevention strategies include targeting the key modifiable factors like hypertension, diabetes, smoking, and high cholesterol. Due to the high cost of our diagnosis system, the majority of our people cannot go for checkups. Nowadays, IoT has unarguably transformed the healthcare industry and is highly beneficial for doctors, and patients. This project proposes a prototype IoT-based Brain Stroke Prediction System by analyzing the key risk factors of stroke and predicting the associated risk status using machine learning technology. Blood glucose level, hypertension status, heart disease status, smoking status, marital status, age, gender, BMI, working type, and residence type are the risk factors employed for this proposed system. The proposed prototype is able to detect blood glucose level which is one of the key risk factors of stroke non-invasively using NIR spectrology. It then analyzes other key risk factors of stroke along with blood glucose level using machine learning technology. The final outcome of this proposed protype is the associative risk status of a person in positive (high) or negative (low) format.

সারসংক্ষেপ

DEVELOPMENT OF A STROKE PREDICTION SYSTEM BASED ON IOT AND MACHINE LEARNING

অনেক এশিয়ান দেশে স্ট্রোক হল অক্ষমতার(disability) অন্যতম প্রধান কারণ, যেখানে নিম্ন ও মধ্যম আয়ের দেশগুলি মৃত্যুর হার বেশি বহন করে। বিশ্বব্যাপী সেরিব্রোভাসকুলার (Cerebrovascular) দুর্ঘটনা (স্ট্রোক) হল মৃত্যুর দ্বিতীয় প্রধান কারণ এবং অক্ষমতার (disability) তৃতীয় প্রধান কারণ, যেখানে বাংলাদেশে এটি মৃত্যুর তৃতীয় প্রধান কারণ। কার্যকর প্রতিরোধের কৌশলগুলির মধ্যে উচ্চ রক্তচাপ, ডায়াবেটিস, ধূমপান এবং উচ্চ কোলেস্টেরলের মতো মূল পরিবর্তনযোগ্য কারণগুলিকে লক্ষ্য করা অন্তর্ভুক্ত। আমাদের ডায়াগনসিস সিস্টেমের উচ্চ খরচের কারণে, আমাদের বেশিরভাগ লোক স্বাস্থ্য পরীক্ষার জন্য যেতে পারে না। আজকাল IoT স্বাস্থ্যসেবা শিল্পকে অবিশ্বাস্যভাবে রূপান্তরিত করেছে এবং ডাক্তার এবং রোগীদের জন্য অত্যন্ত উপকারী। এই প্রকল্পটি (project) একটি প্রোটোটাইপ (prototype) IoT-ভিত্তিক ব্রেন স্ট্রোক পূর্বাভাস (prediction) সিস্টেমের প্রস্তাব করে যা স্ট্রোকের মূল ঝুঁকির কারণগুলি বিশ্লেষণ করে এবং মেশিন লার্নিং (machine learning) প্রযুক্তি ব্যবহার করে সংশ্লিষ্ট ঝুঁকির অবস্থার (risk status)পূর্বাভাস (prediction) দেয়। রক্তের গ্লুকোজের মাত্রা, উচ্চ রক্তচাপের অবস্থা, হৃদরোগের অবস্থা, ধূমপানের অবস্থা, বৈবাহিক অবস্থা, বয়স, লিঙ্গ, BMI, কাজের ধরন এবং বাসস্থানের ধরন এই প্রস্তাবিত সিস্টেমের জন্য ব্যবহৃত ঝুঁকির কারণ। এই প্রস্তাবিত প্রোটোটাইপটি (prototype) রক্তে গ্লুকোজের মাত্রা যা স্ট্রোকের অন্যতম প্রধান ঝুঁকির কারণ সনাক্ত করতে সক্ষম যা এনআইআর বর্ণালী (NIR Spectroscopy) ব্যবহার করে নন-ইনভেসিভেলি (non-invasively) সনাক্ত করতে সক্ষম । তারপর মেশিন লার্নিং প্রযুক্তির মাধ্যমে রক্তের গ্লুকোজের মাত্রা সহ স্ট্রোকের অন্যান্য মূল ঝুঁকির কারণগুলি বিশ্লেষণ করুন। এই প্রস্তাবিত প্রোটাইপের চূড়ান্ত ফলাফল হল ইতিবাচক (উচ্চ) বা নেতিবাচক (নিম্ন) বিন্যাসে একজন ব্যক্তির ঝুঁকির অবস্থা।

ACKNOWLEDGMENT

All praise and glory to Almighty Allah, who provided me with the courage, health, and patience to complete this project work.

I express my sincere gratitude to my project supervisor, Dr. A.K.M. Muzahidul Islam, Professor of United International University, Department of Computer Science & Engineering (CSE). His vast knowledge and wealth of experience have inspired me throughout my academic endeavors and day-to-day existence. His insightful ideas have greatly inspired me and have always led me toward the light when I was having trouble seeing the way. I also express my sincere gratitude to Brig Gen Mohammad Sajjad Hossain, Head of the Department, Department of CSE, MIST whose support was unflinching and very important in the completion of my report and giving it the final shape. He also plays a vital role, either directly or indirectly, in the accomplishment of this project. I would like to convey my thanks to Dr. Muhammad Golam Kibria, Professor, Department of Computer Science and Engineering, University of Liberal Arts Bangladesh, for his kind consent to become my external examiner for this project. I would like to thank Dr. Mahbubur Rahman, Associate Professor, NIVCD (National Institute of CardioVascular Diseases) for guiding me in identifying the key risk factors of stroke.

Finally, I would like to sincerely thank everyone who assisted me in finishing the project. I am also appreciative of all the department employees who have helped, either directly or indirectly. Additionally, I want to thank all of my friends who have supported me during this process. Not to mention, I want to express my gratitude to my parents and other family members for always serving as my role models.

TABLE OF CONTENTS

LISTS OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Introduction

Stroke also known as a cerebrovascular accident (CVA) or brain attack is a medical condition in which poor blood flow to the brain causes cell death Strokes are caused by blocked blood flow to the brain or sudden bleeding in the brain (‘‘About Stroke’’). Not only does the burden of stroke lie in the high mortality but the high morbidity also results in up to 50% of survivors being chronically disabled (Donkor ES, 2018). The impact of stroke extends across individuals, families, healthcare systems, and society as a whole, representing a significant burden. The economic burden of stroke goes beyond direct healthcare expenditures, encompassing indirect costs like lost productivity, disability-adjusted life years (DALYs), and societal expenses (Strilciuc, Stefan et al., 2021). In addressing the cost burden, it is necessary to emphasize preventative strategies. Many things raise the risk of stroke. Some of these risk factors can be changed to help prevent a stroke or future strokes (‘‘Stroke – Causes’’).

In recent years, the integration of machine learning techniques into healthcare has offered promising opportunities for enhancing disease prediction, diagnosis, and treatment. By developing and putting into use a machine learning-based stroke prediction system, this system seeks to advance the field of stroke prevention. This system looks to identify people who are at an increased risk of stroke by utilizing data-driven methods and predictive analytics.

1.2 Motivation

Despite advances in medical science and healthcare delivery, strokes continue to take a heavy toll, underscoring the urgent need for innovative approaches to prevention and early intervention. Low-income nations are far more vulnerable to stroke-related death and disability due to a greater burden of risk factors and a lack of preventive interventions. The elderly are particularly vulnerable; in low- and middle-income countries, those 60 years of age and beyond accounted for 56% of stroke-related DALY loss and 83% of stroke-related deaths. According to a comprehensive review, during the previous forty years, the incidence of stroke rose by more than 100% in low-income nations. (U.K. Saha et al., 2018).

In Bangladesh, non-communicable diseases (NCDs) account for 52% of all causes of

deaths, and ‘CVDs alone account for 27% of those deaths (U.K. Saha et al., 2018). Accurate prediction and timely intervention can mitigate these impacts of stroke. but timely intervention and accurate prediction mean early as well as regular health checkups. But in a developing country like Bangladesh health checkup is treated as an out-of-pocket expenditure. Most of the people here are unwilling to go for a checkup without having any disease due to their financial condition.

This system is meant to be used alongside traditional diagnosis systems which can predict the associative risk status of a person. So that people can take necessary precautions if needed. One of the key benefits of this system is that it is very low-cost and the existing diagnosis systems (diagnosis or medical centers) can incorporate it with their facility. So that people can know their risk factor in a very cost-effective way.

1.3 Objective and the Outcome

The objectives of this project are –

I. To develop IoT based system for collecting one of the key risk factors’ data (blood glucose) using infrared sensors non-invasively.
II. To develop a machine learning model to analyze all the key risk factors’ data and predict the associated risk status.

The outcomes of this project are –

I. Measured blood glucose value from glucose measuring module
II. Predicted possible risk status value of a person

1.4 Organization of the Book

The introductory chapter has described the basic overview, motivation and objective of this project. The rest is organized in the following order -

Chapter-2: Theoretical background and literature review are described in this section.

Chapter-3: Methodology details are discussed.

Chapter-4: Implementation details are depicted in this chapter.

Chapter-5: System result, result analysis, limitation of system and future work are discussed here.

CHAPTER 2
THEORETICAL BACKGROUND

This chapter will cover the detail explanation on the key terms like stroke key factor values, internet of things, machine learning and will also briefly describe similar research works related to proposed system.

2.1 Key Risk Factors of Stroke

There are many risk factors for stroke. Some risk factors for stroke can be changed or managed, while others can't (''Stroke| Johns Hopkins Medicine.''). Risk factors for stroke that can be changed, treated, or medically managed:

- High blood pressure - Blood pressure of 140/90 or higher can damage blood vessels (arteries) that supply blood to the brain. Higher value of blood lead to higher risk of stroke.
- Heart disease - heart disease is the second most important risk factor for stroke, and the major cause of death among survivors of stroke. Heart disease and stroke have many of the same risk factors.
- History of TIAs – TIAs (transient ischemic attacks) are often called mini-strokes. They have the same symptoms as stroke, but the symptoms don't last. If you have had one or more TIAs, you are almost 10 times more likely to have a stroke than someone of the same age and sex who has not had a TIA.
- Diabetes - People with diabetes are at greater risk for a stroke than someone without diabetes.
- Smoking

Risk factors for stroke that can't be changed:

- Older age
- Race
- Gender
- History of prior stroke
- Heredity or genetics

Other risk factors include:

- Where you live.
- Temperature, season, and climate.
- Social and economic factors

2.2 System Concepts

The concept of this system is to analyze the key risk factors of stroke and based on that result make risk status prediction using a machine learning algorithm. The key risk factors used for this system are- blood glucose level, hypertension status, heart disease status, BMI, age, gender, work type, resident type, marital status and smoking habit.

Total system will consist of 2 different module –

1. Module for measuring blood glucose
2. Module for predicting stroke risk factor

2.2.1 Blood Glucose Measuring Module

Here one of the key risk factors blood glucose value is measured non-invasively. Blood glucose level plays as an important key risk factor of stroke. But the traditional blood glucose measuring procedure is invasive, which cause pain and discomfort ness.

In this module blood glucose is measured non-invasively using spectroscopy. Many optical non-invasive glucose detection techniques use an approach that assesses glucose in biological tissue by reflecting, scattering, and transmitting light in accordance to the sample's structure and chemical composition. A photo emitting source like LED or Laser is used as light source and a photodetector is used to receive the reflected light. Then this received value is used for measuring blood glucose level.

Why glucose monitoring?

Blood glucose value of one of the key attribute values of stroke. Hence measuring blood glucose is very important. But the traditional blood glucose measuring method figure tip prick is painful and uncomfortable for a person.

Non-invasive glucose measurement using Near-Infrared (NIR) spectroscopy offers several advantages over traditional invasive methods such as fingerstick testing. NIR spectroscopy allows for glucose measurement without the need for needles or lancets, making it a painless and comfortable alternative for individuals. It also eliminates the risk of infection

associated with invasive methods like fingerstick testing. This is particularly important for individuals with compromised immune systems or those prone to skin infection. It can be performed quickly and easily, without the need for specialized equipment or trained healthcare professionals. By reducing the need for expensive disposable supplies such as lancets and test strips, non-invasive glucose monitoring using NIR spectroscopy has the potential to lower healthcare costs.

2.2.2 Stroke Prediction Module

This module takes input on all the key risk factors – age, gender, hypertension status, heart disease status, blood glucose level, BMI, smoking status, residual status, job type. Using these inputs and based on training data set this machine learning module makes a prediction on associative risk status value. This model is dependent on training dataset. So, training dataset must be select carefully.

2.3 Internet of Things (IOT)

The Internet of Things (IoT) refers to a network of physical devices, vehicles, appliances, and other physical objects that are embedded with sensors, software, and network connectivity, allowing them to collect and share data. The ability of IoT to improve ease and efficiency in many areas of our lives is one of its most important effects. IoT has impacted many different industries and sectors. Examples include smart homes with thermostats, lighting controls, and security cameras that can be operated remotely from smartphones, as well as wearable technology that tracks health indicators and gives immediate feedback. In addition to improving daily conveniences, IoT has also revolutionized industries such as manufacturing, transportation, healthcare, agriculture, and logistics.

2.4 Machine Learning (ML)

In artificial intelligence, machine learning (ML) is the study and creation of statistical algorithms that can learn from data, generalize to new data, and carry out tasks without explicit instructions. Numerous industries, including computer vision, speech recognition, email filtering, natural language processing, agriculture, and medical, have used machine learning techniques. Machine learning algorithms use historical data as input to predict new output values. The way an algorithm learns to make increasingly accurate predictions is a common way to classify traditional machine learning. Supervised learning,

unsupervised learning, semi-supervised learning, and reinforcement learning are the four fundamental methods. A range of algorithms may be employed by data scientists, depending upon the type of data they aim to forecast.

In supervised learning, a model is used to predict the labels on the features of a new dataset after being trained using algorithms to identify patterns in a dataset of features and labels. In unsupervised learning, an algorithm is trained with the data that hasn't been classified or labeled, allowing the algorithm to work on the dataset without any supervision. In this method, the machine's goal is to classify unsorted data based on similarities, patterns, and differences without any prior data training. Clustering and association are the two groups of algorithms that make up unsupervised learning. The goal of Reinforcement Learning is to determine how intelligent agents should behave in a given environment to maximize the concept of cumulative reward.

2.5 Decision Tree

Decision trees is the node-based data structure that are used to test hypotheses with input data. It is a tree-shaped structural graph that begins with a specific choice or inquiry. The input data is compared to the leaf nodes lower in the tree in an attempt to produce the desired, accurate output. They can be made to categorize data in accordance with a schema, and their tree-like form makes them easy to interpret visually. Using decision trees is one method used in the field of supervised learning, or machine learning. The process of training a learning algorithm to create a predictive machine learning model is known as supervised learning.

2.6 Related Works

This section briefly describes our study and analysis of some Existing related works on stroke using IoT devices and machine learning and also works related to non-invasive blood glucose measuring.

Finger pricking is one of the most popular invasive procedures, but it only obtains two to three drops of blood, and those who are checked regularly may become infected and uncomfortable. The devices of the next generation began to be less invasive, requiring the use of tiny skin needles for Continuous Glucose Monitoring (CGM).

Numerous techniques have been developed for improving blood-glucose measurement, and the majority of research papers now show interest in the optical method. It is more

reliable and has an effective cost (Peled, Nina et al., 2002).

The optical method of blood-glucose measurement is the most popular method that has been studied and researched concerning provide a non-invasive measurement. Numerous optical techniques are available for the non-invasive procedure, such as light scattering, polarization technique, photoacoustic spectroscopy, Raman's spectroscopy, near-infrared (NIR), and polarimetry. (Ning et al., 2019, Cole et al., 2019).

Jyoti Yadav et al. (2014) has introduced the glucose sensor. The NIR LED is the basis for this sensor's operation. They have analyzed the glucose concentration using a continuous wave 940 nm spectrum. For the experiment, they used various glucose concentrations. The blood reflectance spectrum was examined during the experiment on a human forearm, and the results showed good accuracy.

The study conducted by Nina Korlina Madzhi et al. (2014) compared the glucose level measurement capabilities of GaAs (950nm), GaAIAs (940nm), and InGaAsP (1450nm) sensors. First, they use test tubes with different percentages of glucose content, then they repeat the process with human blood samples. Compared to 940 nm wavelength, 950 nm has a wider voltage range and a more consistent pattern.

Sandip Panesar et al. (2018) compares machine learning methods with logistic regression for predicting 2-year mortality in hyperacute stroke patients, demonstrating the potential of machine learning techniques for trend delineation, categorization, or prediction.

Chen-Chih Chung et al. (2023) proposes a machine learning-based method using XGBoost for predicting stroke outcomes, highlighting the effectiveness of advanced algorithms in stroke prediction and risk assessment.

Bhattacharya et al. (2019) explores automated diagnosis of stroke using machine learning techniques applied to transcranial Doppler signals, offering insights into novel approaches for stroke detection and diagnosis.

Luo et al. (2016) provides guidelines for developing and reporting machine learning predictive models in biomedical research, offering valuable recommendations for researchers working in the field of machine learning and healthcare.

Asadi et al. (2014) applies machine learning techniques to predict outcomes of acute ischemic stroke following intra-arterial therapy, demonstrating the potential of predictive modeling in stroke management and treatment decision-making.

Nwosu et al. (2020) presents an ensemble machine learning approach to predict stroke

outcomes using electronic health records data, showcasing innovative methods for integrating and analyzing healthcare data for predictive modeling.

Jung et al. (2021) proposes a hybrid model combining random forest and logistic regression for predicting stroke outcomes in patients with atrial fibrillation, demonstrating the potential synergy of different machine learning algorithms for predictive modeling.

Majority of these researches are focusing on continuous vital sign reading. But this proposed model is emphasized on early prediction of a person's health status in a cost-effective way. So that people can take initiative before a major health incident. Also, for low- and middle-income countries like Bangladesh it will be able to provide a very low-cost health checkup facility.

CHAPTER 3

METHODOLOGY

3.1 Proposed Methodology

The whole system is divided into two parts. Firstly, a non-invasive blood glucose measuring module is developed using IoT to measure the blood glucose level of a person which is one of the key risk factors of stroke.

Then this data is feed into the second module for making the system output using all other key attribute values using machine learning algorithm.

Here is the simple system block diagram -

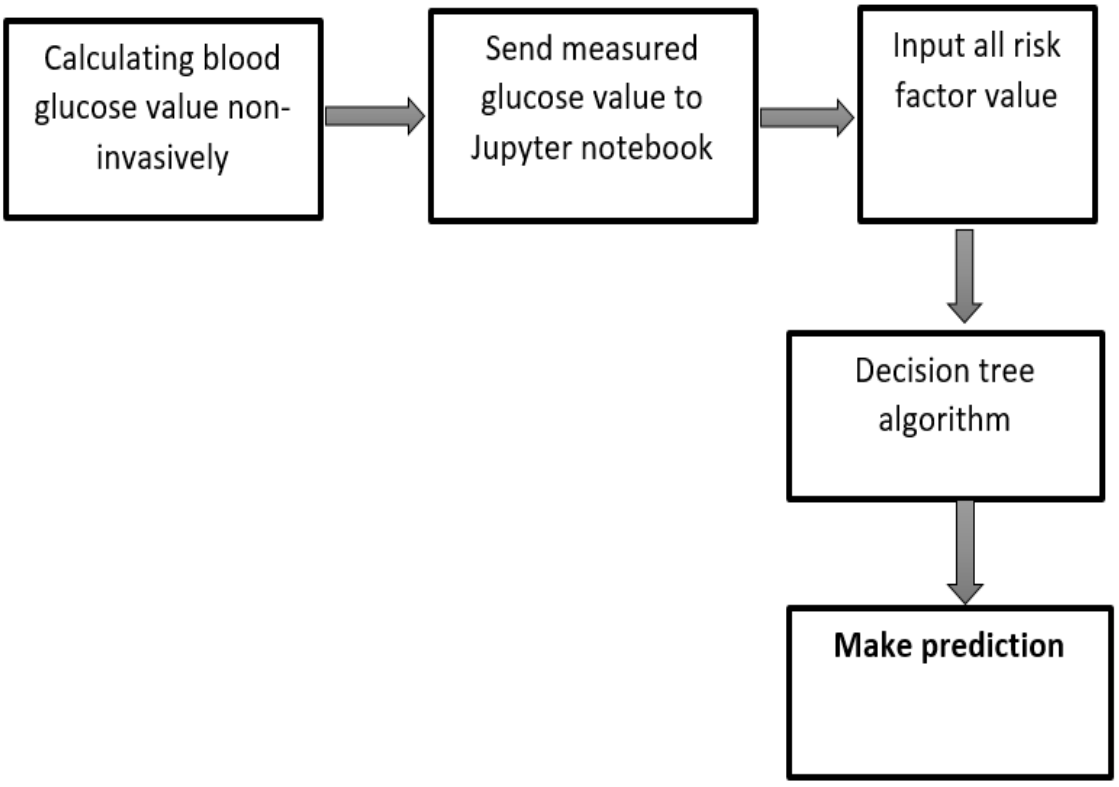


Figure 3.1: System block diagram

3.2 Glucose Measuring Module

This proposed module is for measuring blood glucose level. In this module, glucose value is measured non-invasively using the spectroscopic method by NIR (Near Infrared) spectroscopy. IR (Infrared) light emitter with specific wavelength and photo detector having high sensitivity are used for this purpose. The biological tissues absorb and scatter

light rays that pass through their layers. The mismatch between the refraction index of extracellular fluid and the cell membranes causes light scattering in biological tissues. Changes in blood glucose levels have an impact on how brightly light is reflected off of tissue. By measuring light scattered off and through sample materials, an NIR spectrophotometer makes it possible to quickly and precisely assess their characteristics.

3.2.1 Spectroscopy

Spectroscopy is classified according to the wavelength of the electromagnetic spectrum such as IR spectroscopy, UV spectroscopy and so on (P.S.K. Reddy et al., 2022). In this model, we used NIR spectroscopy.

3.2.2 NIR Spectroscopy

Near-infrared (NIR) spectroscopy is based on the idea that electromagnetic radiation (EM) with wavelengths between 780 and 2,500 nm can be absorbed (P.S.K. Reddy et al., 2022). The transmittance and absorbance of the sample are measured by a detector as light interacts with it. There are some peak points at which glucose absorption is very large. These are 935nm, 1150nm, 1450nm, and 1536nm (Yadav et al., 2014). But at 940 nm wavelength, the attenuation of optical signals by other constituents of the blood like water, platelets, red blood cells, etc. is minimum, hence a desired depth of penetration can be achieved and actual glucose concentration can be predicted (SVajravelu & Kumar, 2013). Hence for this module 940 nm wavelength has used.

3.2.3 Detecting Glucose Value

An IR module of 940 nm wavelength is used for NIR spectrology purpose. This IR module is to be placed on the fingertip of a person. The infrared light emitted from the emitter of the module will be reflected and received by the photodetector of the module. The output current of the photodetector is converted into a voltage signal for preprocessing and then it is filtered and amplified for further use. This amplified signal is fed into ATmega328 microcontroller. The inbuilt ADC block of the microcontroller is used for converting the received analog signal to digital form and used to calculate the glucose value. This converted voltage sends to the Arduino IDE. After making necessary calculation this value is read by machine learning model on Jupyter Notebook. Here is the simple block diagram of this module-

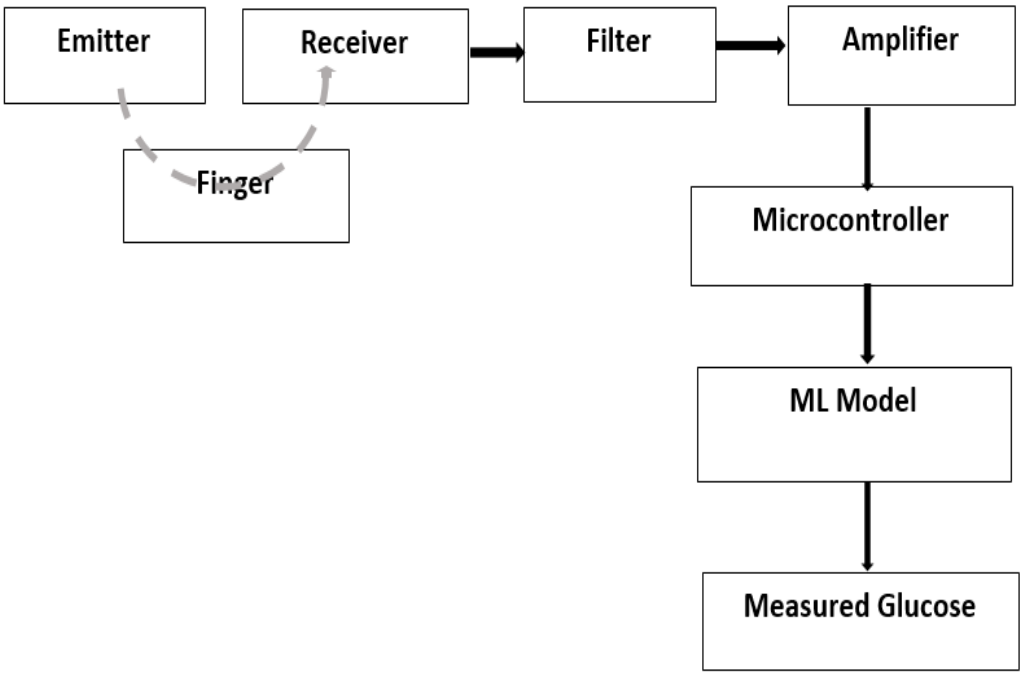


Figure 3.2: Glucose measuring module – block diagram

3.3 Stroke Prediction Module

This module is completely based on machine learning technology. A machine learning application collects all the risk factor values as input. An input form is used to collect all the necessary attribute value. Those data is the preprocessed for further use. Then analyzed and make prediction using machine learning algorithm.

The key factors values used in this model are - Gender, age, hypertension status, heart disease status, glucose value, smoking status, marital status, working type, residence type and BMI. This model takes those inputs and analyze data using Decision Tree Algorithm based on previous input data and make a prediction. The result of the prediction is shown as associate risk status output.

3.4 Data Preparation

Data preparation is the process of preparing raw data so that it is suitable for further processing and analysis. Key steps include –

3.4.1 Data Collection

The first step in data preparation for machine learning is collecting the data you'll need for the model. The sources of this data can vary widely depending on project's requirements.

Data might be pulled from databases, APIs, spreadsheets, or even scrape it from websites. Some projects may also require real-time data streams. Data must be chosen by examining their uniformity, reasonableness, and consistency.

3.4.2 Data Cleaning

For the purpose of developing models, data must be properly prepared and cleansed. The acquired data is placed through a "cleaning" process to make sure it is properly categorized and that any discovered information gaps are filled with the pertinent data. Data cleaning may require –

Handling Missing Values

Missing values happen when certain numerical values are blank in your dataset. Missing data can significantly impede the process of creating predictive models because such values could contain important information that would allow for more precise forecasts. Imputation of missing data is therefore essential. There are numerous ways to deal with missing values, depending on the situation and the available information. some common techniques used for handling missing values are - row deletion, mean/median/mode imputation, creating a prediction model.

Handling Outliers

When data comes from unidentified sources, machine learning algorithms are susceptible to the distribution and range of values. These variables have the potential to degrade both the model's performance and the machine learning training system as a whole. Therefore, it is crucial to identify these abnormalities or outliers using methods like visualization.

Handling Inconsistencies

Inconsistencies in data can throw off analysis and result in misleading information. In the marketing context, this could range from inconsistent naming customer database to conflicting metrics across different analytics platforms. To fix this, domain-specific rules can be applied that standardize naming or metrics to correct these inconsistencies. Data validation techniques can also be helpful here.

3.4.3 Data Transformation

Data transformation is the process of converting cleaned data into a format suitable for machine learning algorithms. This often involves feature scaling and encoding, among other techniques.

Feature Scaling

The technique of normalizing a dataset's feature range is known as feature scaling. Features in real-world datasets frequently vary in terms of magnitude, range, and unit of measuring. Feature scaling must be done in order for machine learning models to interpret these features on the same scale.

Feature Encoding

Categorical values must be converted to numerical format. Feature encoding techniques like one-hot encoding or label encoding can be used to transform these categorical variables into a numeric form that can be fed into machine learning algorithms, though they will still need to be designated and treated as categorical variables for modeling purposes.

3.4.4 Data Reduction

The term data reduction describes the strategies and tactics applied to lower the amount, dimensions, and complexity of data while maintaining its valuable content. In order to make the data more manageable and effective for processing and analysis, it entails removing redundant information, identifying the most pertinent features or patterns from the data, and compressing the data.

3.4.5 Data Splitting

One of the most important things is to fit data into the algorithm and train it to recognize patterns. To forecast the outcome, the model needs to be fed new dataset after it has learned the pattern. To avoid overfitting, two different datasets are not imported for the project's train and test phases. As a result, splitting happens inside of a single dataset. Moreover, it is possible to alter the spit data ratio to obtain improved accuracy.

3.5 Machine Learning Algorithm

Different machine learning algorithm are available to predict the outcome of a system. Depending on the project algorithm can be selected. For predicting the output of the system Decision Tree algorithm is used.

3.5.1 Decision Tree Algorithm

Decision trees are a popular machine learning algorithm for both classification and regression problems.

A decision tree starts with a root node, which does not have any incoming branches. The internal nodes—also referred to as decision nodes—are fed by the outgoing branches that originate from the root node. Both node types evaluate the available attributes to create homogeneous subsets, which are referred to as leaf nodes or terminal nodes. All of the possible outcomes in the dataset are represented by the leaf nodes. Here is a simple structure of a decision tree -

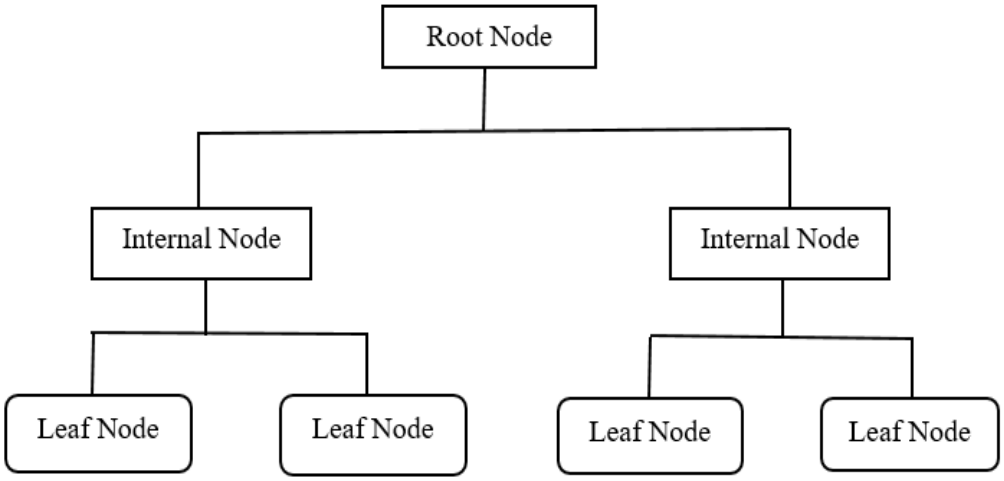


Figure 3.3: Decision Tree

Decision trees are useful for many different kinds of tasks, such as feature selection, regression, and classification. With a Decision Tree, a training model that can be used to forecast the target variable's class or value is constructed by learning simple decision rules based on previous data.

They work with several kinds of data formats and can handle both numerical and categorical data. Different from many other machine learning methods, decision trees do not assume anything about the data distribution. It follows that they don't require any data transformation to be used with any kind of data. Data missing and outliers don't affect decision trees. Accurate forecasts can be made even with noisy data. With multicollinearity, decision trees typically hold up well. Decision trees do not rely on the link between independent variables, in contrast to linear models, where multicollinearity may distort the predicted coefficients and render them untrustworthy. Hence, Decision Tree is used in this proposed system as the machine learning algorithm.

3.6 System Evaluation

3.6.1 Model Evaluation Techniques

The process of evaluating a model using certain metrics to evaluate the model's performance is called model evaluation technique. The two categories of techniques used to evaluate a model's performance are holdout and cross-validation.

Holdout is the easiest strategy. It is utilized in numerous classifiers and neural networks. The dataset is split into train and test datasets for this technique. Typically, the dataset is split into ratios such as 80:20 or 70:30. Typically, a sizable fraction of the dataset is used for model testing, and a larger amount is used for training the model.

In Cross Validation, we do not use the entire dataset for training the model A portion of the dataset is set aside in this strategy to test the model. K Fold Cross Validation is the most widely used type of Cross-Validation among the various types available. To perform K Fold Cross Validation, split the original dataset into k subgroups. Folds are the names for the subsets. This is done k times, with a single fold being used for testing. The model is trained using the remaining k-1 folds. Thus, every data point serves as both the training and test subjects for the model. It is observed that this method lowers the model's error rate and effectively generalizes the model (“Machine Learning”).

3.6.2 Model Evaluation Metrics

To quantify model performance, metrics for evaluation are required. The assessment metrics are selected based on a particular machine learning task (e.g., topic modeling, clustering, regression, ranking, classification, and so on). The evaluation metrics for the classification task are Area Under Curve (AUC), Logarithmic Loss, Confusion Matrix, F-Measure, and Classification Accuracy.

In regression our goal is to predict the continuous values that make up the target variable. To assess the effectiveness of this kind of model, the assessment measures listed below are employed: Mean Absolute Error, Mean Squared Error, Root Mean Square Error, Root Mean Square Logarithmic Error, R2 – Score.

Accuracy

The accuracy function is a simple yet fundamental metric used to evaluate the performance of a classification model, including decision trees. It measures the proportion of correctly predicted instances out of the total instances in the dataset.

Accuracy is calculated using the formula –

Here,

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} = \frac{TP+TN}{TP+TN+FP+FN}$$

True Positives (TP): These are the cases where the model accurately predicts to be positive.

True Negatives (TN): These are the cases where the model accurately predicts to be negative.

False Positives (FP): These are the cases where the model predicts instances as positive, but are negative.

False Negatives (FN): These are the cases where the model predicts instances as negative, but are positive.

Mean Absolute Error (MAE)

Absolute error, as used in machine learning, describes the extent of discrepancy between an observation's true value and its forecast. A set of predictions and observations' average absolute errors is used by MAE to calculate the total group's error magnitude. MAE can also be referred as L1 loss function.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|$$

where, N = total number of data points, yi= actual value, $\hat{y}i$ = predicted value

Root Mean Squared Error (RMSE)

The Root Mean Square Error (RMSE) is the square root of the difference between the given data's expected and actual values. The Root Mean Square Error can be calculated by taking the square root of the MSE. It is the most common metric evolution method applied to regression issues. It is predicated on the idea that errors are unbiased and has a normal distribution. When the RMSE is higher, there are significant differences between the expected and actual values.

The mathematical representation has been shown below:

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n} (y_j - \hat{y}_j)^2}$$

where, N = total number of data points, yi= actual value, ŷi = predicted value.

3.7 Hardware Components

Hardware components are mainly used for IoT module. A brief description of used hardware components is stated in this section. In this proposed prototype max30101 sensor is used for spectroscopic purpose and a microcontroller is used for processing and sending the data to machine learning model.

3.7.1 Microcontroller

In this project, SparkFun RedBoard is used as the microcontroller. It is a microcontroller board based on the Arduino platform, designed to provide an easy-to-use and versatile

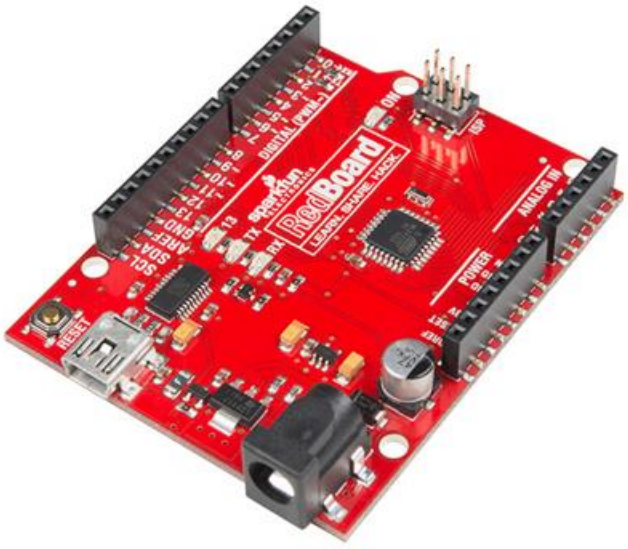


Figure 3.4: SparkFun Redboard (Source: Sparkfun)

development platform for electronics projects. The SparkFun RedBoard is fully compatible with the Arduino development environment. The RedBoard is powered by the ATmega328 microcontroller, the same chip used in the Arduino Uno. This microcontroller offers a good balance of performance and versatility for a wide range of applications. The RedBoard features built-in USB connectivity, allowing it to be easily connected to a computer for programming and serial communication. The RedBoard can be powered via USB or an external power source (e.g., batteries), providing flexibility for different project

requirements. The RedBoard features standard Arduino headers, including digital and analog input/output pins, PWM outputs, UART, SPI, and I2C interfaces. These headers allow users to easily connect sensors, actuators, displays, and other peripherals to the board.

3.7.2 SparkFun Max30101 Sensor

The SparkFun MAX30101 Particle Sensor Breakout is a compact and versatile sensor module designed for measuring a variety of physiological parameters, including heart rate, pulse oximetry (SpO2), and blood oxygen saturation levels. The breakout board is based on the MAX30101 sensor from Maxim Integrated, a highly integrated optical sensor module capable of measuring vital signs and physiological parameters in various applications, with its high-performance sensor, flexible operation modes, and user-friendly design. With built-in photodetector and red, IR, and green LEDs in a single package, the MAX30101 sensor offers programmable operation modes and configurable settings for optimizing performance in different applications and environments. Users can adjust parameters such as LED brightness, sampling rate, and filter settings to suit their specific requirements.

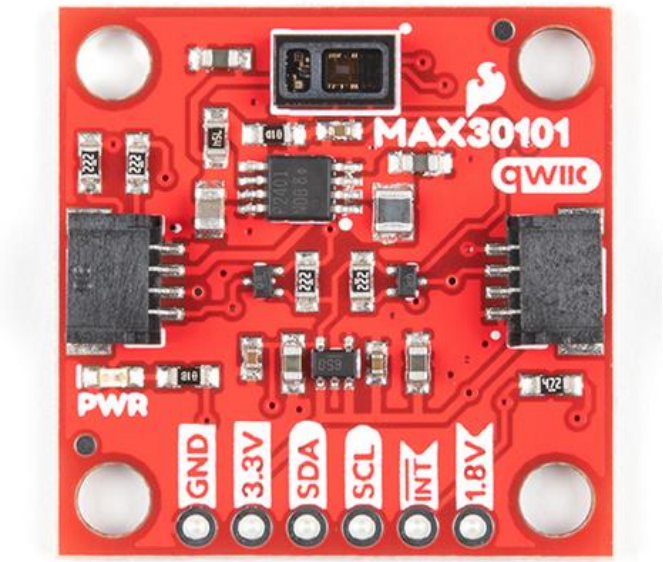


Figure 3.5: max30101 particle sensor (Source: Sparkfun)

Here is the pin configuration of max30101 particle sensor -

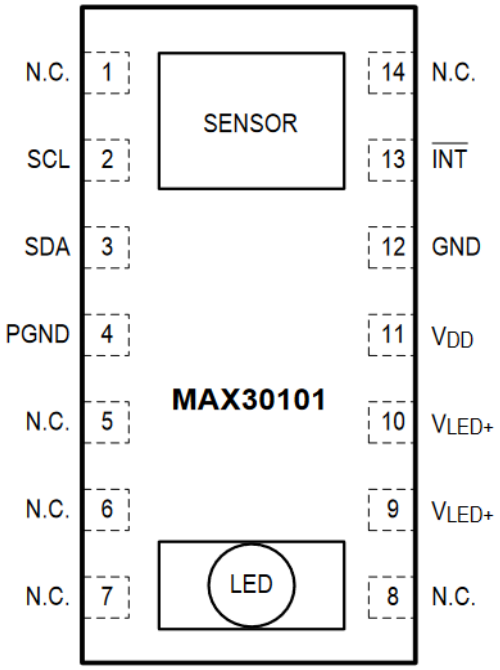


Figure 3.6: Max30101 pin diagram (Source: Sparkfun)

3.8 Software Components

As for software components Arduino IDE is used in the glucose measuring module and Jupyter Notebook is used booth for glucose measuring and stroke prediction module.

3.7.1 Arduino IDE

The Arduino Integrated Development Environment (IDE) is an open-source software application that provides a platform for writing, compiling, and uploading code to Arduino microcontroller boards. The IDE features a user-friendly interface designed to be simple and intuitive, making it easy for beginners to get started with programming Arduino boards. It includes built-in tools and features for writing, editing, and managing code, as well as compiling and uploading sketches to Arduino boards. Arduino code, known as sketches, is written using the Arduino programming language (based on C/C++), which is specifically tailored for programming Arduino boards. The Arduino IDE comes with a built-in library manager that allows users to easily install and manage libraries of pre-written code (called sketches) for various sensors, actuators, communication protocols, and other peripherals.

The Arduino IDE provides a beginner-friendly and versatile platform for programming Arduino boards, enabling users to bring their ideas to life through code and create a wide range of interactive projects and prototypes.

3.7.2 Jupyter NoteBook

Jupyter Notebook is used as a machine-learning platform in this project. It is an open-source web application to create and share documents containing live code, equations, visualizations, and narrative text. It provides an interactive computing environment that supports multiple programming languages, including Python, R, Julia, and others. Users can write and execute code directly within the notebook interface, making it ideal for data analysis, scientific computing, machine learning, and other computational tasks. Jupyter Notebook integrates seamlessly with popular data science libraries and tools, such as NumPy, pandas, Matplotlib, scikit-learn, TensorFlow, and PyTorch. Users can import these libraries and leverage their functionality directly within the notebook environment to perform data manipulation, analysis, visualization, and modeling tasks. It provides a flexible and powerful platform for interactive computing, data analysis, and scientific research, empowering users to explore, experiment, and communicate their findings effectively through a combination of code, text, and visualizations.

3.7.3 Programming Languages

C Language

The C programming language, developed by Dennis Ritchie at Bell Labs in the early 1970s, stands as a cornerstone of modern computing. Renowned for its simplicity, efficiency, and portability, C provides developers with a powerful toolset for system programming, embedded systems, and performance-critical applications. Its straightforward syntax, rich standard library, and support for low-level features like pointers and manual memory management have made it a favorite among programmers seeking fine-grained control over their code. With its modular structure and ability to run on diverse hardware platforms, C remains an essential language in fields ranging from operating systems and game development to high-performance computing. Its influence can be felt across a spectrum of modern programming languages, cementing its status as a foundational technology in the world of software development.

Python

Python, created by Guido van Rossum and first released in 1991, has emerged as one of the most popular and versatile programming languages in the world. Known for its simplicity, readability, and ease of use, Python emphasizes code readability and simplicity, making it an ideal choice for beginners and experienced programmers alike. Python's extensive standard library and comprehensive ecosystem of third-party libraries empower developers to build a wide range of applications, from web development and data analysis to artificial intelligence and scientific computing. Its interpreted nature and dynamic typing facilitate rapid development and prototyping, while its cross-platform compatibility ensures that Python code can run seamlessly on various operating systems. With a vibrant community, robust documentation, and a commitment to open-source principles, Python continues to evolve and thrive as a leading language in the ever-expanding landscape of technology and software development.

CHAPTER 4
IMPLEMENTATION

This section will provide a concise explanation of the entire project implementation procedure. It will be explained how the different quality data were gathered, preprocessed, analyzed, and used for this project. Additionally, the hardware connections that link the machine learning module to the Internet of Things will be explained.

4.2 Blood Glucose Measuring Module

Hardware components are used only for glucose measuring module. For blood glucose measuring module sensor is connected to the microcontroller board using jumper wires. The microcontroller is connected to the Arduino IDE using a USB cable. Pin connections are –

- Pin GND of max30101 is connected to pin GND of Redboard
- Pin 3.3V max30101 is connected to the pin 5V of Redboard
- Pin SDA of max30101 is connected to pin SDA of Redboard
- Pin SCL of max30101 is connected to pin SCL of Redboard

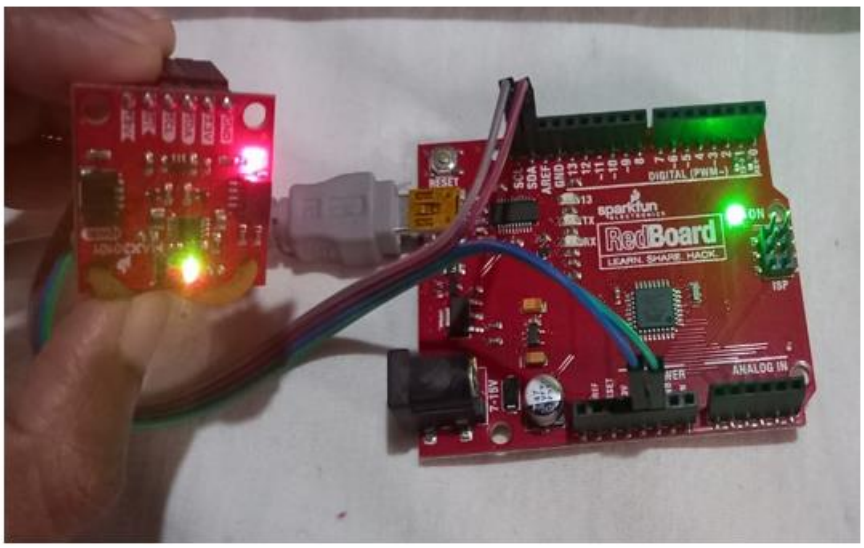


Figure 4.1: Sensor assembly

4.2.1 Data Collecting and Preprocessing

Using max30101 particle sensor is collected from the finger of a person. A finger is placed on the sensor. Light is emitted from the IR LED of the sensor. The intensity of the reflected light is received by the photodetector of the sensor.

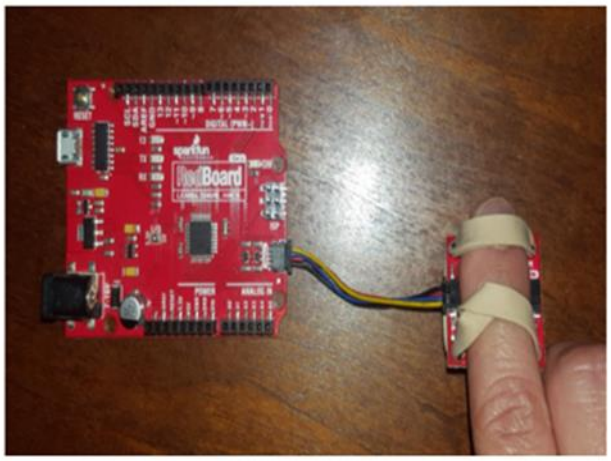


Figure 4.2: Collecting data using sensor

```
23  void loop()
24  {
25    long irValue = particleSensor.getIR();
26    if (irValue >threshold && flag==0) {
27
28      //Serial.print("Value: ");
29      for (i = 0; i < sampleSize; i++) {
30        //Serial.println(irValue);
31        irAvg += irValue;
32        delay(10);              // Small delay between samples if needed
33      }
34      irAvg /= sampleSize;
35      Serial.println(irAvg);
36      flag=1;
37    if (i == sampleSize) {
38          // Stop the loop
39          while (1);
40        }
41    }
42    delay(1000);
```

Figure 4.3: Arduino code for reading sensor data

Received data at the photodiode is filtered and amplified for further processing using sensor-prebuilt module.

Then it is collected by the Arduino IDE using its pre-built function. For measuring the blood glucose value correctly 20 samples is collected. Then an average is taken out of those values for maintaining right accuracy. The average value is printed on serial monitor using println() function of Arduino IDE.

4.2.2 Training Dataset for Glucose Measuring

Here is the sample training dataset used to train the machine learning model to calculate glucose value. It associates measured average voltage value with glucose level. Here second column contains the average voltage value and third column represents corresponding glucose level.

SI	Voltage level	Glucose value
1	82320	86
2	82407	77
3	82284	91
4	82349	60
5	82367	93
6	82335	94
7	82284	91
8	82209	97
9	32485	72
10	82355	57
11	82345	59
12	82385	57
13	77701	112
14	74978	108
15	82330	69
16	82365	76
17	82425	94
18	82340	90
19	82438	69
20	82430	77

Table 4.1: Training dataset for glucose measuring

4.2.3 Measuring Blood Glucose

The data collected and preprocessed on Arduino IDE is received by the Jupyter Notebook using **serial.read()** function. The received data is decoded and then used for measuring the glucose value.

```
import serial
import time
import numpy as np
from sklearn import svm

ser = serial.Serial('COM5', 9600)
time.sleep(1)
data = ser.readline().decode().strip()
ser.close()
print(data)
```

```
82438
```

Figure 4.5: Python code for reading serial data from sensor

4.3 Stroke Prediction Module

This module entirely depends on the machine learning algorithm and quality of training dataset. The machine learning model is first trained with training dataset and then used to predict.

4.3.1 Training Dataset for Stroke Prediction

Various training datasets are available to train a machine learning model on different machine learning websites. For this project, I have used a training data set from Kaggle.com to train the system for predicting the associative risk status of a person. The key fields of this dataset support the key factors of stroke perfectly as suggested by the doctor. The attributes that used in this dataset are age, gender, hypertension status, heart disease status, BMI, smoking status, blood glucose value, working type and resident type. Here is the sample of the dataset –

id	gender	age	Hypertension	Heart disease	Ever married	Work type	Residence type	avg_glucose level	bmi	Smoking status	stroke
1	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
2	Female	61	0	0	Yes	Self-employed	Rural	202.21	N/A	never smoked	1
3	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
4	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
5	Female	79	1	0	Yes	Self-employed	Rural	174.12	24	never smoked	1
6	Male	81	0	0	Yes	Private	Urban	186.21	29	formerly smoked	1
7	Male	74	1	1	Yes	Private	Rural	70.09	27.4	never smoked	1
8	Female	69	0	0	No	Private	Urban	94.39	22.8	never smoked	1
9	Female	59	0	0	Yes	Private	Rural	76.15	N/A	Unknown	1
10	Female	78	0	0	Yes	Private	Urban	58.57	24.2	Unknown	1
11	Female	29	0	0	Yes	Private	Rural	60.74	20	never smoked	0
12	Male	36	0	0	No	Private	Rural	233.52	40.9	never smoked	0
13	Female	19	0	0	No	Private	Urban	110.7	38.5	never smoked	0
14	Female	41	0	0	Yes	Govt_job	Rural	78.93	30.9	formerly smoked	0
15	Female	23	0	0	No	Private	Urban	124.5	33.4	Unknown	0
16	Male	14	0	0	No	children	Urban	57.95	17.1	Unknown	0
17	Male	35	0	0	Yes	Private	Rural	92.82	28.6	Unknown	0
18	Male	45	0	0	Yes	Private	Rural	58.25	24	smokes	0
19	Female	52	1	0	Yes	Private	Rural	213.54	32	never smoked	0
20	Male	19	0	0	No	Private	Urban	74.86	28.4	never smoked	0

Table 4.1: Training dataset for stroke Prediction

4.3.2 Preprocessing Training Dataset

The training data set contains different types of values for attributes. Some attribute values are in numeric format like age, BMI, glucose value and some are in character format (categorical data) like – hypertension status, marital status etc. There are also some missing and NULL values. So, the dataset needs preprocessing. The attributes with character type values have been converted into numeric format. And also, the null and missing values have been handled.

Removing unwanted value – for this dataset serial number column are not necessary. It is important to remove unwanted data or column. Here is the code for removing unnecessary data –

```
stroke_dataset = pd.read_csv('stroke.csv')

# droping unwanted column
stroke_dataset.drop('id', axis = 1, inplace=True)
```

Figure 4.6: Removing unwanted data

Handling categorical data – categorical data values need to convert into numerical value before using the dataset. Code for handling categorical values –

Handling gender column value –

```
# handling gender col
def change(col):
    if col == 'Male':
        return 0
    elif col =='Female':
        return 1
    else:
        return 2

stroke_dataset['gender'] = stroke_dataset['gender'].apply(change)
```

Figure 4.7: Handling categorical data1

Handling working type column data –

```
# handling work_type
def alter(col):
    if col == 'Private':
        return 0
    elif col == 'Self-employed':
        return 1
    elif col == 'Govt_job':
        return 2
    elif col == 'children':
        return 3
    else:
        return 4

stroke_dataset['work_type'] = stroke_dataset['work_type'].apply(alter)
```

Figure 4.8: Handling categorical data2

Handling smoking status and bmi column values –

```
# handling smoking_status
def change1(col):
    if col == 'formerly smoked':
        return 0
    elif col == 'never smoked':
        return 1
    elif col == 'smokes':
        return 2
    else:
        return 3

stroke_dataset['smoking_status'] = stroke_dataset['smoking_status'].apply(change1)

# handling BMI null values
stroke_dataset['bmi'] = stroke_dataset['bmi'].fillna(stroke_dataset['bmi'].median())
```

Figure 4.9: Handling categorical data 3

4.4 Combining the Entire System

After collecting all the required data and necessary preprocessing, input data is used for making the prediction.

CHAPTER 5
RESULT AND DISCUSSION

This section shows the result for the Glucose Measuring Module and the system result. This section also shows the system evaluation.

5.1 Glucose Measuring Module

After receiving the average reflected voltage level from the fingertip and doing the necessary processing on that value, this module makes result based on previous training data. Glucose value is shown in mg/dL (milligrams per decilitre) unit.

```
In [2]: #take input from arduino
        import serial
        import time
        import numpy as np
        from sklearn import svm
        import pandas as pd
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import accuracy_score

        glucose_dataset = pd.read_csv('glucose.csv')
        glucose_dataset.drop('SI', axis = 1, inplace=True)

        X = glucose_dataset.drop(columns = 'glucose')
        y = glucose_dataset['glucose']
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

        model = DecisionTreeClassifier()
        model.fit(X_train.values, y_train)

        ser = serial.Serial('COM5', 9600)
        time.sleep(1)
        data = ser.readline().decode().strip()
        #print(data)
        ser.close()

        prediction = model.predict([[data]])
        prediction

Out[2]: array([124], dtype=int64)
```

Figure 5.1: Glucose module output

The result from the glucometer and the value from the planned module show a strong association. The mapping chart of glucose concentration measured by the intended system and simultaneously for the same individual using a glucometer is displayed in the following figure.

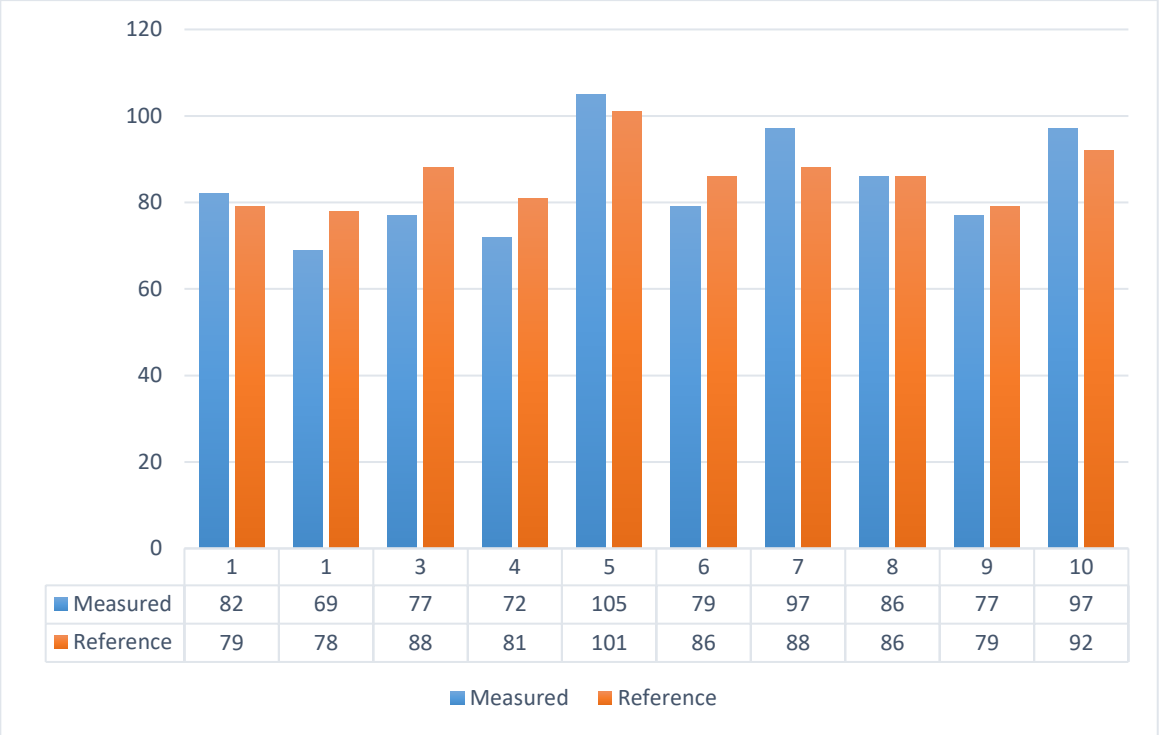


Figure 5.2: Measured and reference glucose concentration

5.2 Stroke Prediction Module

After receiving the stroke attribute values as input Decision Tree algorithm makes the associative risk status prediction based on the training data set. Prediction value 0 (zero) means associative risk status is LOW and value 1 (one) means associative risk status is high. Based on the prediction the result is shown in text format.

Here is the input form for key factor values gender-female, age-35, hypertension status - no hypertension, heart disease status – no heart disease, marital staus-1, working type- govt. job, glucose value- 98 mg/dL, BMI-20 and smoking status- never smoke.

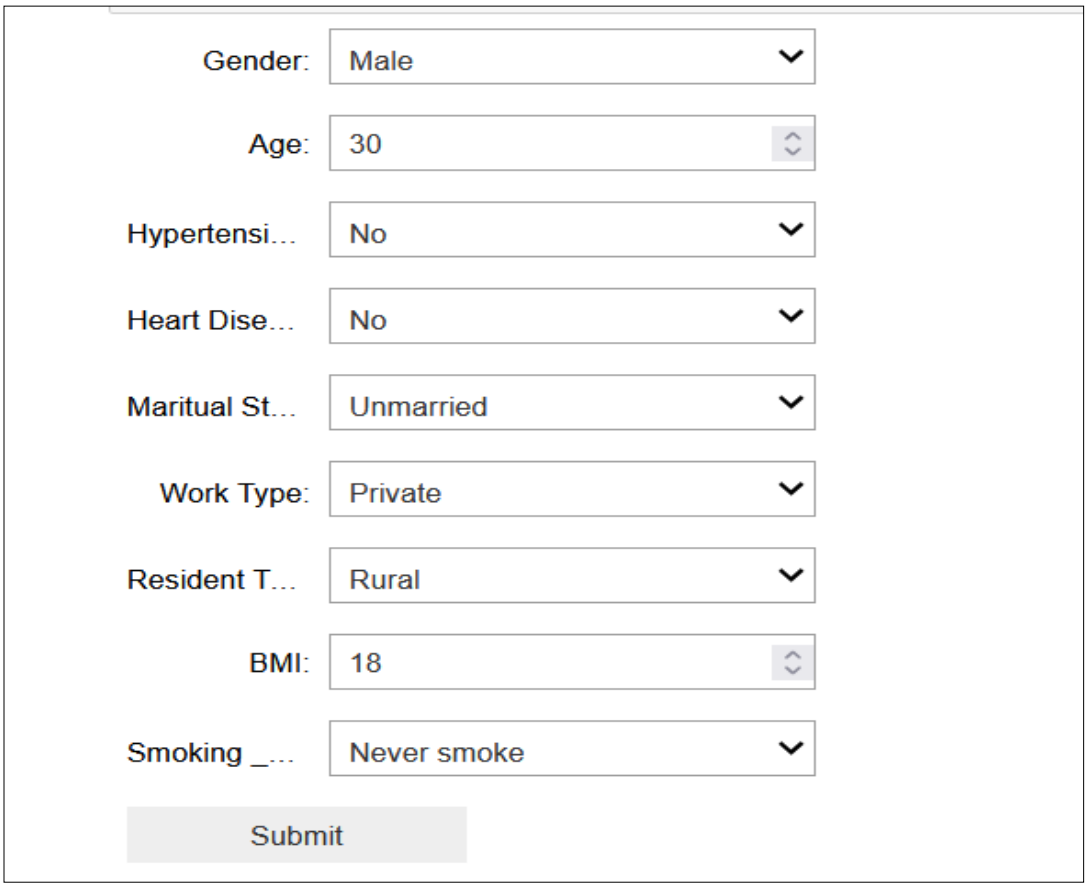


Figure 5.3: Input form1

After analyzing these input value and also the measured glucose value the system output is as follow –

```
if(prediction== 0):
    print("Your associative risk status is Negative. Have a Good Day !")
else:
    print("Your Associative risk status is HIGH. please maintain your daily life.")
```

```
Your associative risk status is Negative. Have a Good Day !
```

Figure 5.4: System prediction 1

Here is the input form for key factor values gender- male, age- 80, hypertension status - no hypertension, heart disease status – yes, marital status- unmarried, working type- private

job, glucose value- 224 mg/dL, BMI-33 and smoking status- formerly smoked -

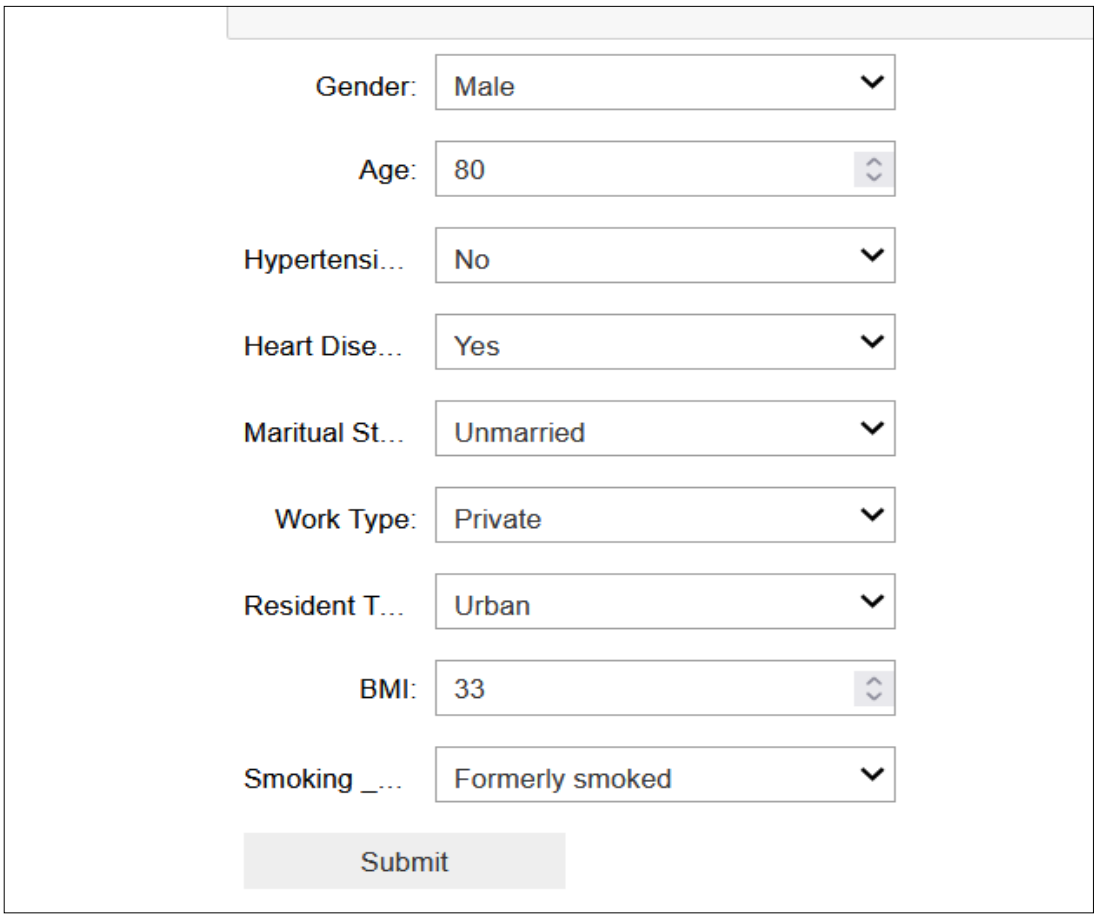


Figure 5.5: Input form 2

Here is the system output for these input values –

```
model = DecisionTreeClassifier()
model.fit(X,y)

array = [1, 80, 0, 1, 0, 0, 0, 225, 36, 0]

prediction = model.predict([array])

if(prediction== 0):
    print("Your associative risk status is Negative. Have a Good Day !")
else:
    print("Your Associative risk status is HIGH. please maintain your daily life.")
```

```
Your Associative risk status is HIGH. please maintain your daily life.
```

Figure 5.6: System output 2

5.3 System Evaluation

As briefly mentioned in the previous chapter, the model is trained with the set of data using a 10-fold cross-validation approach that dynamically selects the training and testing with fixed proportion each time. Here is the number of test and training data available for each fold –

```
In [4]: kf =KFold(n_splits=10, shuffle=True, random_state=42)
        cnt = 1
        for train_index, test_index in kf.split(X, y):
            print(f'Fold:{cnt}, Train set: {len(train_index)}, Test set:{len(test_index)}')
            cnt += 1

        Fold:1, Train set: 4599, Test set:511
        Fold:2, Train set: 4599, Test set:511
        Fold:3, Train set: 4599, Test set:511
        Fold:4, Train set: 4599, Test set:511
        Fold:5, Train set: 4599, Test set:511
        Fold:6, Train set: 4599, Test set:511
        Fold:7, Train set: 4599, Test set:511
        Fold:8, Train set: 4599, Test set:511
        Fold:9, Train set: 4599, Test set:511
        Fold:10, Train set: 4599, Test set:511
```

Figure 5.7: System cross validation set

Here is the score for each fold –

```
score = cross_val_score(tree.DecisionTreeClassifier(random_state= 42), X, y, cv= kf, scoring="accuracy")
print(f'Scores for each fold are: {score}')
print(f'Average score: {"{:.2f}".format(score.mean())}')

Scores for each fold are: [0.89823875 0.88845401 0.91193738 0.92367906 0.91780822 0.91585127
 0.90215264 0.88454012 0.91585127 0.92172211]
Average score: 0.91
```

Figure 5.8: Cross validation score

As described in previous chapter the system accuracy is evaluated by Accuracy, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) evaluation metrices.

The evaluation value with these evaluation matrices is-

```
prediction = model.predict(X_test)
accuracy = accuracy_score(y_test, prediction)
print("Accuracy:", accuracy)

# Calculate MAE
mae = mean_absolute_error(y_test, prediction)
print("Mean Absolute Error:", mae)

# Calculate MSE
mse = mean_squared_error(y_test, prediction)
#print("Mean Squared Error:", mse)

# Calculate RMSE
RMSE = math.sqrt(mse)
print("Root Mean Square Error:", RMSE)
```

```
Accuracy: 0.913894324853229
Mean Absolute Error: 0.08610567514677103
Root Mean Square Error: 0.2934376852873043
```

Figure 5.9: System Evaluation score

5.4 System Benefits

This system can produce the output instantly. So, a person can see the result immediately. As for the glucose measuring module, it is non-invasive so people won't feel pain or discomfort while giving sample. The entire system is very easy to use, and low-cost. So, people can see the result at a very low charge. Thus, it can reduce their medical checkup burden.

5.5 Limitations

The entire system is dependent on how the machine learning model is trained. If the data is incomplete, noisy, or biased, it can negatively impact the model's performance and generalization ability.

This model can only make predictions instantly, and cannot store the prediction value for future use.

The IoT module is able to take input only one key attribute value.

5.6 Future Work

For future work, more attributes value could be taken automatically using sensors, the system could store data for future use. Person’s data could be stored under their Id/ account so it can be used to predict based on previous data. Output could be included comment on each attribute value.

REFERESCES

Centers for Disease Control and Prevention, "About Stroke|cdc.gov,". [Online]. Available: https://www.cdc.gov/stroke/about.htm/. Accessed: Mar. 10, 2024.

National Heart, Lung and Blood Institute, "Stroke - Causes and Risk Factors | NHLBI, NIH,". [Online]. Available: https://www.nhlbi.nih.gov/health/stroke/causes/. Accessed: Mar. 10, 2024.

Johns Hopkins Medicine, "Stroke | Johns Hopkins Medicine,". [Online]. Available: https://www.hopkinsmedicine.org/health/conditions-and-diseases/stroke/. Accessed: Mar. 10, 2024.

E. S. Donkor, "Stroke in the 21st Century: A Snapshot of the Burden, Epidemiology, and Quality of Life," Stroke Research and Treatment, vol. 2018, p. 3238165, 2018. [Online]. Available: doi:10.1155/2018/3238165.

S. Strilciuc, D. A. Grad, C. Radu, D. Chira, A. Stan, M. Ungureanu, A. Gheorghe, and F. D. Muresanu, "The economic burden of stroke: a systematic review of cost of illness studies," Journal of Medicine and Life, vol. 14, no. 5, pp. 606–619, 2021. doi: 10.25122/jml-2021-0361.

U. K. Saha, M. B. Alam, A. K. M. F. Rahman, A. H. M. E. Hussain, S. R. Mashreky, G. Mandal, and Q. D. Mohammad, "Epidemiology of stroke: findings from a community-based survey in rural Bangladesh," Public Health, vol. 160, pp. 26-32, 2018. [Online]. Available: doi: 10.1016/j.puhe.2018.03.024.

N. Peled, D. Wong, and S. L. Gwalani, "Comparison of glucose levels in capillary blood samples obtained from a variety of body sites," Diabetes Technol. Ther., vol. 4, no. 1, pp. 35-44, 2002. [Online]. Available: doi: 10.1089/15209150252924067.

E.H. Yoo and S.Y. Lee, "Glucose biosensors: an overview of use in clinical practice," Sensors (Basel), vol. 10, no. 5, pp. 4558-4576, 2010. [online]. Available: doi: 10.3390/s100504558,

P. S. K. Reddy, D. Mahesh, C. U. Teja, M. Janaki, and K. Mannem, "Non-Invasive Glucose Monitoring Using NIR Spectroscopy," in Journal of Physics: Conference Series, vol. 2325, no. 1, p. 012021, Aug. 2022.

M. J. Goetz, G. L. Coté, R. Erckens, W. March, and M. Motamedi, "Application of a multivariate technique to Raman spectra for quantification of body chemicals," IEEE Trans. Biomed. Eng., vol. 42, no. 7, pp. 728-731, 1995. [Online]. Available: doi: 10.1109/10.391172.

J. Yadav, A. Rani, V. Singh, and B. M. Murari, "Near-infrared LED based non-invasive blood glucose sensor," in 2014 International Conference on Signal Processing and Integrated Networks (SPIN), pp. 591-594, IEEE, Feb. 2014.

N. K. Madzhi, S. A. Shamsuddin, and M. F. Abdullah, "Comparative investigation using GaAs (950nm), GaAIAs (940nm) and InGaAsP (1450nm) sensors for development of non-invasive optical blood glucose measurement system," in 2014 IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), pp. 1-6, IEEE, Nov. 2014.

A. Vajravelu and N. Kumar, "Determination of blood glucose concentration by using wavelet transform and neural networks," Iran. J. Med. Sci., vol. 38, no. 1, pp. 51-56, Mar. 2013

E. J. Benjamin et al., "Heart Disease and Stroke Statistics-2018 Update: A Report From the American Heart Association," Circulation, vol. 137, no. 12, pp. e67–e492, 2018. [Online]. Available: doi: 10.1161/CIR.0000000000000558.

S. Panesar, R. D'Souza, F.-C. Yeh, and J. Fernandez-Miranda, "Machine learning versus logistic regression methods for 2-year mortality prognostication in a small, heterogeneous glioma database," World Neurosurgery, vol. 2, 2019. [Online]. Available: doi: 10.1016/j.wnsx.2019.100012.

C. C. Chung, E. C. Su, J. H. Chen, Y. T. Chen, and C. Y. Kuo, "XGBoost-based simple three-item model accurately predicts outcomes of acute ischemic stroke," Diagnostics, vol. 13, no. 5, pp. 842, 2023. [Online]. Available: doi: 10.3390/diagnostics13050842.

M. I. U. Zaman, S. Tabassum, M. S. Ullah, A. Rahaman, S. Nahar, and A. K. M. Muzahidul Islam, "Towards IoT and ML driven cardiac status prediction system," in 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), May 2019, pp. 1–6.

Ramli, Dzati & Ghazali, Najah & Tay, Lina, “Ischemic Stroke Detection System with Computer Aided Diagnostic Capability,” Procedia Computer Science, vol 126, pp. 393-402, 2018. [Online]. Available: doi: 10.1016/j.procs.2018.07.273.

W. Luo et al., "Guidelines for developing and reporting machine learning predictive models in biomedical research: a multidisciplinary view," Journal of medical Internet research, vol. 18, no. 12, pp. e323, 2016. [Online]. Available: doi: 10.2196/jmir.5870.

H. Asadi, R. Dowling, B. Yan, and P. Mitchell, "Machine learning for outcome prediction of acute ischemic stroke post intra-arterial therapy," PloS One, vol. 9, no. 2, pp. e88225, 2014. [Online]. Available: doi: 10.1371/journal.pone.0088225.

C. Nwosu, S. Dev, P. Bhardwaj, B. Veeravalli, and D. John, "Predicting stroke from electronic health records," in Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE Engineering in Medicine and Biology Society. Conference, 2019, pp. 5704-5707. [Online]. Available: https://doi.org/10.1109/EMBC.2019.8857234.

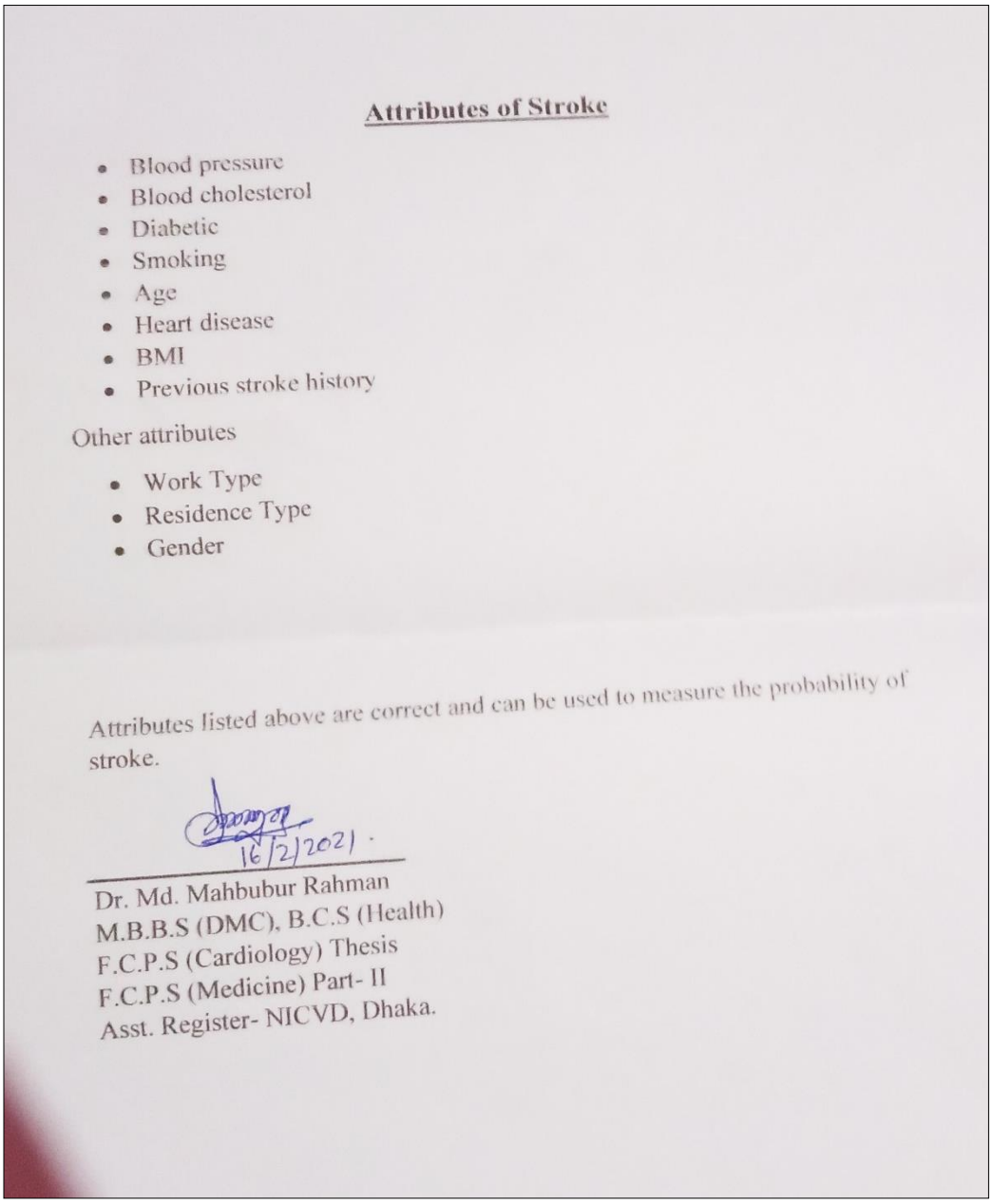
S. Jung, M. K. Song, E. Lee, S. Bae, Y. Y. Kim, D. Lee, M. J. Lee, and S. Yoo, "Predicting ischemic stroke in patients with atrial fibrillation using machine learning," Front. Biosci. (Landmark Ed.), vol. 27, no. 3, p. 80, 2022. [Online]. Available: https://doi.org/10.31083/j.fbl2703080.

GeeksforGeeks, “Machine Learning Model Evaluation – Geeks for Geeks.”. [Online]. Available: https://www.geeksforgeeks.org/machine-learning-model-evaluation/. Accessed: Mar. 10, 2024.

Sparkfun, “www.sparkfun.com” Accessed: Mar. 12, 2024.

APPENDIX A

Doctor's agreement on key risk factors of stroke -



Attributes of Stroke

- Blood pressure
- Blood cholesterol
- Diabetic
- Smoking
- Age
- Heart disease
- BMI
- Previous stroke history

Other attributes

- Work Type
- Residence Type
- Gender

Attributes listed above are correct and can be used to measure the probability of stroke.

16/2/2021.

Dr. Md. Mahbubur Rahman
M.B.B.S (DMC), B.C.S (Health)
F.C.P.S (Cardiology) Thesis
F.C.P.S (Medicine) Part- II
Asst. Register- NICVD, Dhaka.