An Expert System for Insulin Dosage Prediction

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE
OF
MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted by:

Amandeep Prasad
2K21/CSE/04

Under the supervision of
Prof. Shailender Kumar

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

JUNE, 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi – 110042

CANDIDATE'S DECLARATION

I, Amandeep Prasad, Roll No. 2K21/CSE/04 student of M. Tech (Computer Science and Engineering), hereby declare that the project Dissertation "An Expert System for Insulin Dosage Prediction" which is submitted by me to the Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of and Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi   Amandeep Prasad

Date: 2K21/CSE/04

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi – 110042

CERTIFICATE

I hereby certify that the Project Dissertation titled **An Expert System for Insulin Dosage Prediction** which is submitted by Amandeep Prasad, Roll No. 2K21/CSE/04, Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi **Prof. SHAILENDER KUMAR**

Date: Department of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi – 110042

ACKNOWLEDGMENT

The success of this project depends on the help and contribution of a large number of people as well as the organization. I am grateful to everyone who contributed to the project's success. I'd want to convey my gratitude to **Prof. SHAILENDER KUMAR**, my project guide, for allowing me to work on this research under his supervision. His unwavering support and encouragement have taught me that the process of learning is more important than the ultimate result. Throughout all of the progress reviews, I am appreciative to the panel faculty for their assistance, ongoing monitoring, and motivation to complete my project. They assisted me with fresh ideas, gave crucial information, and motivated me to finish the task.

AMANDEEP PRASAD

2K21/CSE/04

ABSTRACT

Diabetes Mellitus is a chronic metabolic disorder. Blood glucose levels (BGLs) should be properly adjusted so that diabetic patients can live normal lives without the danger of developing major issues over time. However, for a variety of reasons, the majority of diabetic patients' blood glucose levels are not under good control. Although it's necessary for diabetes patients to regulate their BGLs to follow the standard preventative measures like eating well and exercising, the most critical step in the healing process is taking the right dosage of insulin. In this paper, we propose an artificial neural network (ANN) based approach to forecast the ideal insulin dosage required for the diabetic patient. Several patients' data sets containing various characteristics, including weight, fast blood sugar, and gender, were used to train and test the suggested model. The suggested model performed well in determining the right dosage of insulin.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ML	Machine Learning
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short Tern Memory
RFA	Random Forest Algorithm
BP	Back Propagation
DL	Deep Learning
RNN	Recurrent Neural Networks
API	Application Programming Interface
CSV	Comma Separated Value

CHAPTER 1

INTRODUCTION

Diabetes mellitus, sometimes referred to as diabetes, is a series of metabolic illnesses characterised by high blood sugar levels that are brought on by issues with insulin production, action, or both. Type 1 and type 2 diabetes are the two subtypes[4]. Only 5–10% of diabetics have type 1 diabetes, which is characterised by a total absence of insulin and is brought on by the immune system killing insulin-producing cells in the pancreas. Conversely, Type 2 diabetes is brought on by a confluence of inadequate compensatory insulin production and insulin resistance, affecting 90–95% of patients and being more prevalent in social settings3]. Regardless of the primary criterion he adheres to, many diabetes board frameworks have been developed to aid patients in self-management of their illnesses [3]. Intricate information-driven strategies for creating precise expectation models of how glucose is metabolised are hence the main focus of current research [1].

Machine learning algorithms can be used to help diabetes patients decide on their insulin dosage by analysing patient data and offering personalised suggestions based on individual patient characteristics. One approach involves applying supervised learning techniques to develop a model using information from a sizable cohort of diabetes patients. The insulin dosages for new patients can thus be predicted using the factors of their age, weight, blood sugar levels, and insulin sensitivity [21].

Another approach is to gradually modify insulin dosages in response to patient feedback using reinforcement learning algorithms. This requires adjusting insulin dosages in response to changes in patient blood sugar levels in order to maintain adequate glucose levels while lowering the risk of hypoglycaemia. Additionally, patterns and trends in patient data can be discovered using machine learning approaches, such as changes in insulin sensitivity over time, which helps medical personnel make better decisions about changing insulin dosage.

Machine learning methods generally have the potential to significantly improve the precision and accuracy of insulin administration for diabetic patients, leading to better health outcomes and a higher quality of life [5].

In addition to the methods I've discussed, customised models for insulin administration can be made using machine learning approaches. These models take into account the distinct patient characteristics including age, weight, body mass index, insulin resistance, and other medical issues to provide individualised recommendations for insulin dose.

For instance, one study used machine learning to develop a customised model for predicting the dosages of insulin in persons with type 2 diabetes. The algorithm took into account a variety of patient parameters, including fasting blood glucose levels, haemoglobin A1c levels, body mass index, and medications, to produce personalised insulin dose recommendations [13]. The researchers found that their algorithm was capable of accurately estimating insulin dosages for brand-new patients, with an average prediction error of less than 10%.

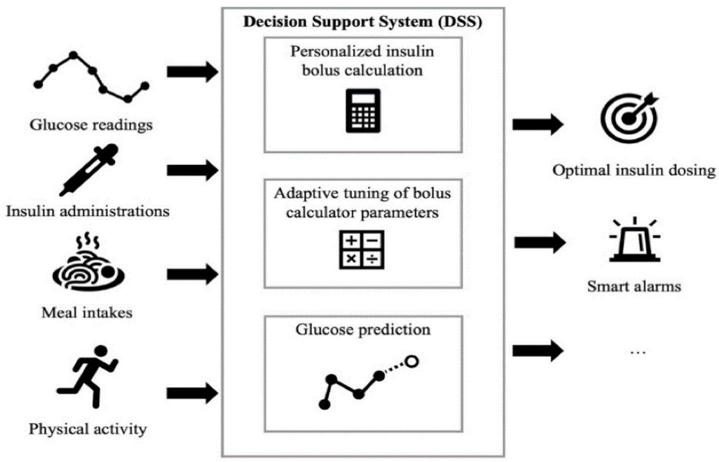

Fig 1.1: Optimal Insulin Dosing flow

Machine learning techniques have a lot of potential for detecting and adjusting insulin dosage for diabetics. Machine learning algorithms can offer tailored recommendations for insulin administration that are more precise and efficient than conventional approaches by analysing massive datasets of patient data and taking into account unique patient features. Machine learning algorithms can also find patterns and trends in patient data, which can help with insulin dose recommendations and improve the health of diabetics. It is possible that we will see more cutting-edge uses of these techniques in diabetes treatment as machine learning technology develops, significantly enhancing patient outcomes and quality of life.

CHAPTER 2

LITERATURE SURVEY

Diabetes, a chronic illness characterized by elevated blood sugar levels, is managed in large part by insulin. Diabetes patients either produce insufficient insulin (those with Type 1 diabetes) or are unable to use the insulin they do make efficiently (Type 2 diabetes). Insulin controls the metabolism of glucose, enabling cells to take up sugar from the bloodstream and use it as fuel. Without enough insulin, blood glucose levels rise, which can cause a number of problems [6].

For diabetics to control their blood sugar levels and avoid complications, maintaining proper insulin levels is essential. However, figuring out the ideal insulin dosage might be difficult. It necessitates careful consideration of a number of variables, including personal traits, way of life, food, and physical exercise. Both hyperglycaemia which is also referred as high blood sugar and hypoglycaemia known as low blood sugar can have major health repercussions if insulin dosage is incorrect [6].

In predicting the appropriate insulin dosage for diabetic patients, machine learning techniques have showed potential. Machine learning algorithms can find patterns and associations by using previous patient data, which includes blood sugar levels, insulin dosages, diet information, and other pertinent variables [19]. Then, these algorithms can produce prediction models to calculate the ideal insulin dosage for a certain set of circumstances.

The benefit of machine learning models is their capacity to take several variables into account concurrently and adjust to the demands of specific patients.

They can examine large datasets and find hidden patterns that conventional analysis might miss. These models can enhance their accuracy over time and offer individualized insulin dosage recommendations by continuously learning from new data [6].

By easing the load on patients and healthcare professionals, machine learning-based insulin dosage prediction systems have the potential to improve diabetes management. They can help to enhance overall glycemic control, optimize insulin therapy, and reduce the risk of some major future complications. It is important to remember that such models should be created and verified using rigorous scientific procedures, taking into consideration a variety of patient groups and taking into account the individual differences in insulin response [13].

Although machine learning shows promise for predicting insulin dosage, it is critical to incorporate these models within a thorough diabetes care strategy. Effective diabetes management continues to depend on regular blood sugar testing, constant collaboration with medical professionals, and adherence to individualized treatment programs.

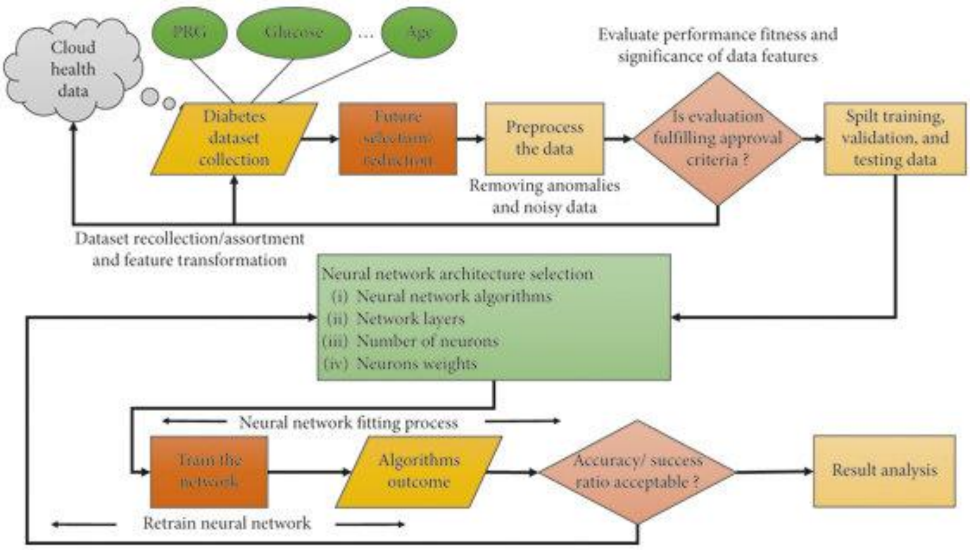

Fig. 2.1: Basic architecture of a machine learning model.

- “Diabetes Prediction using Machine Learning Algorithms”

  Aishwarya Mujumdar , V Vaidehi Dr.

The material covers how hospitals currently detect and treat diabetes as well as how persons with diabetes are at high risk for a number of ailments. The paper emphasises the important part big data analytics may play in the healthcare industry because it can examine enormous databases to unearth hidden patterns and facts. The research suggests a diabetes prediction model that takes into account extrinsic factors that cause diabetes in addition to regular components like glucose, BMI, age, and insulin because the accuracy of the current technique for diabetes categorization and prediction is low. The new dataset raises classification precision, and a pipeline approach is suggested to raise diabetes classification precision [12].

- “Predicting Diabetes Mellitus with Machine Learning Techniques”

  Quan Zou, Kaiyang Qu,Yamei Luo

The paper discusses the chronic condition known as diabetes mellitus, which is characterised by elevated blood sugar levels, as well as any possible consequences. Machine learning has been used in medicine to predict diabetes mellitus due to the disease's rising occurrence. Using data from a hospital in Luzhou, China, this study employs decision tree, random forest, and neural network approaches to predict diabetes [13]. Following five times of cross-validation, the best models were chosen for independent test experiments. To minimise dimensionality, they used PCA and mRMR. When all characteristics were included, the findings revealed that random forest had the greatest prediction accuracy (ACC = 0.8084) [13].

- “Predicting Inpatient Glucose Levels and Insulin Dosing by Machine Learning on Electronic Health Records”

  Xiran Liu, BS1 , Ivana Jankovic, MD2 , Jonathan H Chen

The study used electronic health record data and supervised machine learning techniques to create predictive models that could assist inpatient insulin management. In order to minimise major morbidity and death in hospitalised patients, it was vital to keep glucose levels within a desired range. Although individual BGL levels and insulin injection dosage can be highly unpredictable and challenging to predict with accuracy, Here the study has also found that after using an accurate predictive machine learning method for insulin dosage can help alter the dosage in a daily basis[14]. For forecasting the average daily glucose levels, the models exhibited a mean absolute error of 21 mg/dL, an R-squared value of 0.4, and a sensitivity of Predicting whether a patient's blood glucose levels will be more than the intended range the next day has a sensitivity of 0.73 and a specificity of 0.79 [14].

- “Smart Technique of Insulin Dose Prediction for Type-1 Diabetic Patients”

K. S. Kathe, U. A. Deshpande

In this work, a control method is suggested for keeping type 1 diabetes patients' blood glucose levels within the normal range. The system is made up of a Wireless and enhanced Body Area Network (WBAN) and a controller device with it, that reads blood glucose and insulin dosage, takes the size of the meal as input, calculates the maximum glucose level, and forecasts the amount of insulin that will be needed [15]. Using an insulin pump, the patient is given the prescribed amount of insulin. The proposed approach is intended for people with type-1 diabetes who wish to self-monitor and automatically regulate their blood glucose levels [15].

- “Machine learning for initial insulin estimation in hospitalized patients”

Minho Nuyen, Ivna Jakovic, Lauynas Kolinskis, Michael Balochi, and Jonathan H Chen

This study set out to determine if machine learning can estimate the first total daily dosage (TDD) of insulin for inpatients more accurately than the currently suggested guidelines-based dosing method. The study examined the electronic health records of 16,848 inpatients who administered subcutaneous insulin and had blood glucose levels that were controlled to within the target range of 100 to 180 mg/dL on a given day between 2008 and 2020. A group of machine learning techniques, including regularised regression, random forest, and gradient boosted tree models, were created to anticipate TDD [16].

The results showed that the machine learning model outperformed traditional clinical calculators in predicting TDD, with a mean absolute percent dosage prediction error of 51% for patients requiring more than 6 units.

- “Comparison Data Mining Techniques to Prediction Diabetes Mellitus”

  Aswan Supriyadi Sunge

Diabetes is a chronic ailment that is characterised by elevated glucose levels. Diabetes is predicted and analysed using a variety of automated techniques. One data mining approach may help with a patient's ailment diagnosis. Before a patient becomes ill, expectations may help prevent disorder and save lives [19]. Choosing a legal categorization enhances the system's accuracy and veracity as levels are advanced. The majority of diabetics are not aware of their risk factors before being diagnosed. Using information from the dataset, this approach creates five prediction models with nine input variables and one output variable. This study looked at the ability of Naive Bayes, Decision Tree, SVM, K-NN, and ANN models to predict diabetes mellitus.

The goal of (Yang et al., 2018) is to use deep learning techniques to forecast pharmaceutical formulations. Machine learning models are created by (Murphree et al., 2018) to forecast which patients who were first prescribed metformin would achieve and maintain control of their blood glucose after a year of treatment. Four machine learning methods are suggested by (Veh et al., 2019) as a solution to the issue of safety in diabetes management: (1) grammatical evolution for mid-term continuous blood glucose level prediction (2) support vector computers to forecast hypoglycemic episodes during meal times Data mining is used to profile diabetes management situations, artificial neural networks are used to forecast hypoglycaemic episodes overnight, and so forth. Machine learning approaches are used in the (Han et al., 2019) study to forecast the physical stability of solid dispersions. Machine learning approaches were used to create a unique prediction model for the physical stability of solid dispersion formulations. A machine learning technique is presented by Balasooriya et al. (2020) to create a short-term prediction model utilising non-invasive data, such as prior glucose levels (measured or previously predicted), medicine dose, food intake, and lifestyles of diabetes patients. Focus on a total of 36,652 eligible individuals from the Henan Rural Cohort Study in order to investigate the effectiveness of machine learning algorithms for predicting risk of type 2 diabetes mellitus (T2DM) in a rural Chinese population (Zhang et al., 2020). To imitate the metabolic behaviour of physiological blood glucose techniques, (Munoz-Organero, 2020) design, build, validate, and compare a new hybrid model that decomposes a deep machine learning model. The issue of AdaBoost producing several decision stumps was resolved by the suggested stacking strategy, which combined the intelligent models and improved model performance (Kalagotla et. al., 2021). Other significant works include (Gupta et. al., 2021).

CHAPTER 3

COMPARATIVE ANALYSIS

AUTHOR	STUDY OBJECTIVE	MACHINE LEARNING METHODS USED	PROS	CONS
Yang et al., 2018	Predict pharmaceutical formulations	DL methods were used	Highly accurate forecasts and the potential to speed up and lower the cost of drug development	A scarcity of facts and difficulty generalising to novel forms
Murphree et al., 2018	Predict glycaemic control in patients taking metformin	ML models were used	Individualized forecasts have the potential to enhance patient outcomes.	Only includes metformin medication, and sample size is modest.
Vehí et al., 2019	Improve safety in diabetes management	Grammatical evolution	Multiple strategies to address various facets of managing diabetes, potential to	Potential for overfitting, limited to specific diabetes care subproblems

			lower hypoglycemia	
Han et al., 2019	Predict physical stability of solid dispersions	ML techniques were used	Enhanced prediction effectiveness and efficiency; ability to speed up and cut costs associated with medication development	Restricted to limited datasets and the physical stability of solid dispersions
Balasooriya et al., 2020	Predict short-term glucose levels in diabetes patients	ML algorithm using non-invasive data	Individualized predictions, non-invasive method, and potential to lessen hypoglycemia	Short-term projections only, with a chance of overfitting
Zhang et al., 2020	Targeting only a specific population for getting results	Machine learning model	Predicting T2DM risk with high accuracy and the potential to enhance population health outcomes	Limited to rural Chinese population and potential for bias
Dave et al., 2020	Predict hypoglycaemia	Machine learning model	High degree of predictability and potential	CGM data only, with a chance of overfitting

	in diabetes patients	using CGM data	for lowering hypoglycemia	
Muñoz-organero, 2020	Mimic physiological blood glucose methods using machine learning	Hybrid model using stacking technique	Enhanced model performance and reduced risk of hypoglycemia	Restricted to a certain methodology and subject to overfitting
Kalagotla et al., 2021	Improve prediction accuracy in diabetes management	Adaboost with decision tree classifiers	High predictability of glycaemic states and potential to lower hypoglycemia	Restricted to a certain methodology and subject to restricted to a certain population and subject to bias overfitting
Gupta et al., 2021	Predict complications of diabetes using electronic health records	Machine learning models	Potential to enhance patient outcomes and prediction accuracy	Limited to specific population and potential for bias

CHAPTER 4

SYSTEM ANALYSIS

4.1 Existing System

Insulin may enable blood glucose to enter cells and be utilised as fuel. As a consequence, blood glucose levels continue to fall within a predetermined range. Chronic diseases like diabetes have the potential to expedite a catastrophe for world health. But it is challenging for doctors to establish an early diagnosis of diabetes due to the complicated connections between several factors. Diabetes has an impact on several human organs, including the foot, kidney, eye, heart, and nerves.

4.1.1 Disadvantages

The complex interactions between many factors, however, make it difficult for medical practitioners to predict diabetes early.

4.2 Proposed System

De-individualized expectation models are used in this study because they are more generalizable and practical because they don't have to worry about physiological cutoff thresholds or do any actual work besides obtaining EHR information.
The Random Forest Regression Algorithm can forecast the insulin dosage if diabetes is identified by the ANN algorithm. Fake Brain Organizations can forecast diabetes. The two calculations are produced using datasets from PIMA and UCI.

4.2.1 Advantages:

Due to the large range of individual variability in blood glucose levels and insulin dosages, it is impossible to make precise forecasts. It is more prudent to plan ahead if you know the patient's blood glucose level will be high. because the normal 24-hour blood glucose level may be predicted.

4.3 System Study

4.3.1 Artificial neural network (ANN):

A form of machine learning algorithm known as an artificial neural network (ANN) mimics the structure and operation of organic neurons in the human brain. Complex issues requiring pattern recognition, categorization, and prediction can be resolved with ANNs.

Information is processed and sent by layers of artificial neurons found in ANNs. Each neuron gets information from other neurons or the outside environment, processes it internally, and then sends out an output. Weights are applied to the connections between neurons, and these weights are adjusted throughout training to improve the functioning of the network.

ANNs are frequently used in machine learning to perform tasks including audio and picture recognition, natural language processing, and predictive modelling. ANNs have demonstrated to be effective in jobs that call for discovering intricate patterns in huge datasets. supervised learning, unsupervised learning, or a combination of the two can be used to train ANNs [9].

In supervised learning, the network is trained using samples that have labels and whose input and associated output are known. By changing the weights of the connections between neurons, the network gains the ability to translate inputs into outputs. Unsupervised learning is the process of teaching a network to recognise patterns and structures in unlabeled data without the use of explicit instructions.

All things considered, ANNs are a powerful machine learning tool that enable sophisticated problem solving and pattern detection. Predictive modelling, speech recognition, natural language processing, computer vision, and other fields all use them.

A system of mathematical equations is used in artificial neural networks (ANNs) to enable learning and prediction. The following are some basic equations used in ANNs:

1. Weighted Sum: The output of each neuron in a layer is determined by adding a bias term to the weighted sum of the inputs, followed by an activation function. This is illustrative of:

   $z = \sum (w * x) + b$

When b is the bias term, w stands for the weights, x for the inputs, and z for the weighted sum.

2. The activation function is used to create the final output of the neuron by applying it to the weighted sum's output. The sigmoid function, ReLU function, and softmax function are the three most often utilised activation functions.

   $a = f(z)$

   Where f is the activation function.

3. Backpropagation is a learning algorithm that is used to modify the weights and biases of the network while it is being trained. The parameters of the network must be updated after computing the gradient of the loss function with respect to those parameters. The following is a representation of the weights update rule:

   $w' = w - \alpha * \partial L/\partial w$

   Where L is the loss function, L/w is the gradient of the loss function with respect to weight w, and w' is the updated weight.

4. Loss Function: The discrepancy between the expected and actual outputs is measured using the loss function. The mean squared error and the cross-entropy loss function are the two most often utilised loss functions.

   $L = f(y, \hat{y})$

where y is the output itself, The projected result is, and the loss function is f.

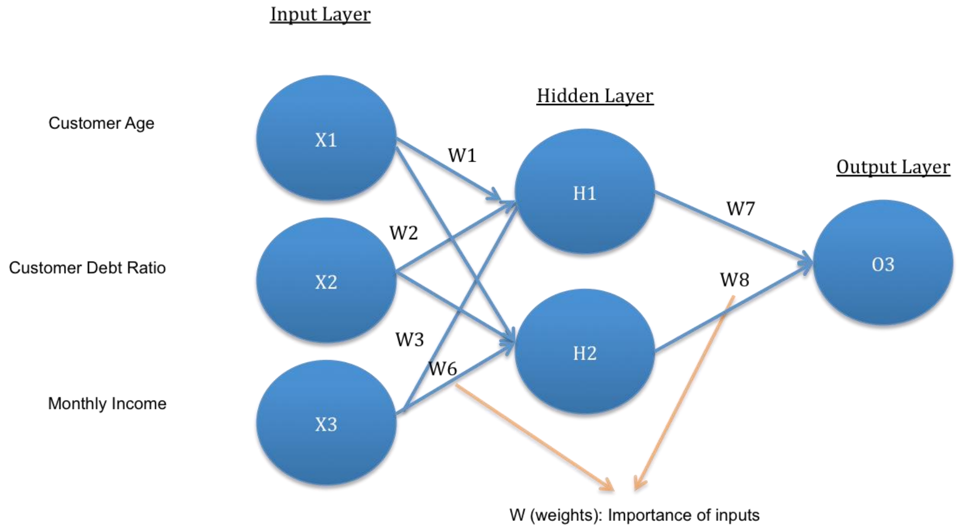

Fig. 4.1: Artificial Neural Network Flow

4.3.2 Random Forest Algorithm

Random forest is a machine learning method that is applied to classification, regression, and other tasks. Using a randomly random subset of the training data and characteristics, the approach creates a forest of decision trees. The outcome is then determined by the forest's overall production. [21].

A thorough explanation of the random forest algorithm is provided below:

1. Random subspace method: Rather than using all of the features, the algorithm randomly chooses a subset of features from the training data at each node in the decision tree. As a result, the accuracy of the forest's trees is increased while the correlation between them is decreased.

2. Bootstrap aggregating (bagging): Using replacement sampling, the approach creates several bootstrap samples from the training data and then trains a decision tree on each sample. As a result, the model has a lower variance and is less susceptible to data noise and outliers.

3. Voting: To arrive at the final conclusion, the algorithm combines the predictions of each tree in the forest. While in regression issues the mean of the projected values is calculated, in classification problems the majority vote of the trees is obtained.

4. Approach tweaking for hyperparameters: The algorithm enables adjustment for hyperparameters such the number of trees in the forest, the maximum depth of the trees, and the minimum number of samples needed to divide a node. Finding the ideal values that reduce the model's error can be done using methods like grid search or randomised search.

5. Out-of-bag (OOB) error estimation: Using OOB samples that were excluded from each decision tree's training, the technique calculates how well the model will perform on fresh data. Through this, the model's error can be quickly and objectively estimated without the use of cross-validation.

With its high dimensionality and ability to handle big datasets with non-linear correlations between the features and the target variable, random forest is a potent method. Additionally, it can manage missing values and data outliers and is resistant to overfitting. Numerous applications, such as bioinformatics, economics, and image recognition, have made extensive use of random forests [28].

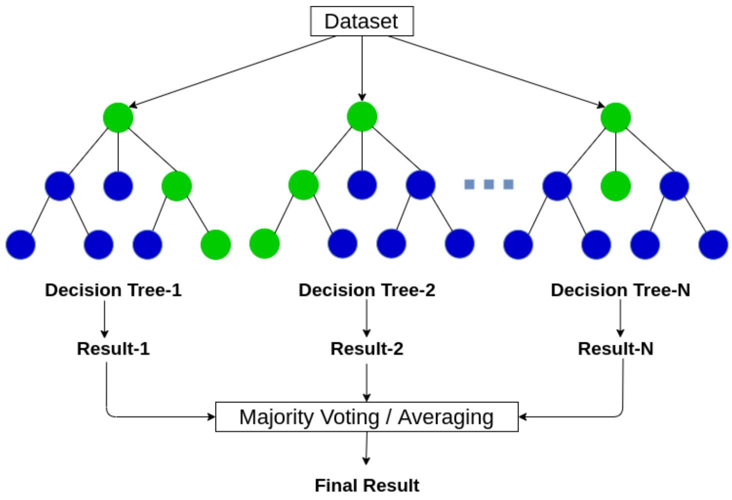

Fig 4.2: Random Forest Workflow

The random forest algorithm is described by the following equations:

1. Random subspace method: A random subset of characteristics is chosen for each node of the decision tree. Let m be the number of features chosen for a particular node, and let M be the overall number of features. The formula for calculating the likelihood of choosing a specific feature is:

   $p = m/M$

2. Bootstrap aggregating (bagging): N samples are randomly chosen with replacement from the training data to create a bootstrap sample [12]. The formula for calculating the likelihood of choosing a specific sample is

   $p = 1/n$

3. Voting: To arrive at the final conclusion, the algorithm combines the predictions of each tree in the forest. The majority vote of the trees is used to classify problems. Let K be the total number of classes, and let y hat i represent the predicted class for the i-th tree. The final prognosis is then made by:

   k = 1 to K, where y hat = argmax(sum(1(y hat i = k))

   The mean of the anticipated values is computed in regression problems. Let T be the total number of trees and y hat i be the anticipated value for the i-th tree. The final prognosis is then made by:

   for I = 1 to T, y hat = (1/T) * sum(y hat i)

4. Hyperparameter turning: Turning the hyperparameters includes changing the number of trees in the forest (T), the maximum depth of the trees (d), and the minimum number of samples required to split a node (s). These hyperparameters can be changed using techniques like grid search or randomised search to lessen the model's error.

5. Out-of-bag (OOB) error estimation: The approach makes advantage of the OOB samples—samples that were not used in the training of each decision tree—to predict how well the model performs on fresh data. For classification problems, the OOB error is determined as the percentage of incorrectly classified samples, and for regression issues, as the mean squared error.

4.4 System Requirements

4.4.1 Hardware Requirements

Depending on the particular requirements of your project, building a machine learning model may require a range of hardware components. Here are some fundamental hardware specifications to think about:

1. CPU: Your computer's central processing unit (CPU) is in charge of carrying out instructions. For machine learning applications, a top-tier CPU with several cores and fast clock speeds is preferred.
2. GPU: By parallelizing computations, graphics processing units (GPUs) can dramatically speed up machine learning processes. The best dedicated GPU for deep learning model training is one with a lot of CUDA cores and VRAM.
3. RAM: The temporary storage area where the computer keeps data that is currently in use is known as random access memory (RAM). Large datasets can be processed by your computer more quickly the more RAM it has.
4. Storage: Given that machine learning datasets have a tendency to get quite large, you will require a sizable quantity of storage capacity to save them. Because it offers higher read/write rates than a conventional hard disc drive (HDD), a solid-state drive (SSD) is advised.
5. Power supply: When executing demanding computations for extended periods of time, a high-quality power supply is essential to supplying reliable power to your components.
6. Cooling is crucial to keep your components from overheating because performing heavy computations can produce a lot of heat. A top-notch CPU cooler and case fans can keep your system operating at peak efficiency.
7. Motherboard: The motherboard links and facilitates communication between all of your components. Select a motherboard that has enough expansion slots and ports for your needs, is compatible with your CPU and GPU, and meets your other requirements.

Remember that these are merely minimum system requirements and that larger or more complicated machine learning tasks may call for more capable hardware.

Hardware Component	Recommended Specification
CPU	Intel Core i7 or i9 with multiple cores and high clock speeds
GPU	NVIDIA GeForce or Quadro with at least 8GB VRAM and a large number of CUDA cores
RAM	At least 16GB, but preferably 32GB or more
Storage	Solid-state drive (SSD) with at least 500GB capacity
Power Supply	High-quality 600W or greater power supply
Cooling	High-quality CPU cooler and case fans
Motherboard	Compatible with CPU and GPU, with sufficient expansion slots and ports

Fig 4.3: Hardware Requirements

4.4.2 Software Requirements

A machine learning model needs specialised software tools and libraries in addition to the necessary hardware. The following are some fundamental software prerequisites for creating a machine learning model:

1. Operating system: A current operating system, such as Windows 10, macOS, or Linux, is required. Check the documentation for each tool you intend to use because some machine learning frameworks and libraries may have special OS requirements.
2. Python: Python provides a large selection of libraries and frameworks and is a well-known programming language for machine learning. Install the most recent version of Python and ensure sure all relevant packages, such as NumPy, Pandas, and Matplotlib, are installed.
3. Frameworks for machine learning: Popular frameworks for machine learning include TensorFlow, PyTorch, and Scikit-learn. Pick a framework based on your requirements and level of experience.

4. Integrated Development Environment (IDE): An IDE is a piece of software that offers a complete setting for coding, debugging, and testing. PyCharm, Spyder, and Jupyter Notebook are a few well-known IDEs for machine learning.
5. Version control is crucial for keeping track of code changes and cooperating with other developers. The machine learning community uses Git, which is the most well-liked version control programme.
6. Data visualisation solutions can help you more effectively visualise and interpret your data. Examples include Tableau and Power BI.

Cloud computing services can supply the necessary processing power for many machine learning projects because they can be computationally intensive. Examples of these services are Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure.

These are only a few examples of the software specifications needed to create a machine learning model. Depending on your project's needs and preferences, you'll need particular tools and libraries.

Software	Recommended Tools/Libraries
Operating System	Windows 10, macOS, or Linux
Programming Language	Python 3.x with NumPy, Pandas, Matplotlib, and other necessary packages
Machine Learning Frameworks	TensorFlow, PyTorch, Scikit-learn, Keras, and others
Integrated Development Environment (IDE)	PyCharm, Spyder, Jupyter Notebook, and others
Version Control	Git
Data Visualization Tools	Tableau, Power BI, Matplotlib, Seaborn, and others
Cloud Computing Services	Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure, and others

Fig 4.4: Software Requirements

4.5 System Study

4.5.1 4.5.1 Feasibility Study

The project's sensibility is assessed at this point, and a fundamental understanding is equipped with a crucial task plan and a few cost projections. During the design assessment, investigating the reasonableness of the proposed structure is central. This will ensure that the association will not face any issues as a result of the recommended strategy. It is essential to have a comprehensive comprehension of the fundamental requirements for the structure of the feasibility study.

The following three aspects have a total impact on the attainability study:

- At the same time, take into account social, technological, and economic viability.

4.5.2 Economic Feasibility

This assessment is being done to see what kind of money related impact the improvement will have on the alliance. The group only has a limited amount of resources at its disposal to contribute to the system's creative effort. To keep track of the costs, a money order ought to be used. The constructed system was within the budget because a significant portion of the improvement was public space. Only those specific items should have been purchased.

4.5.3 Technical Feasibility

The specific requirements or reachability of the structure are the focus of this evaluation. The mechanical assets that can be gotten to ought not be troubled a lot by any developed design. How much mechanical assets accessible will be vivaciously stressed in accordance with this. In this way, the client will defy certain guidelines. Because of the way that its execution just expects near zero changes, the developed construction should have a low interest.

4.5.4 Social Feasibility

The client's level of familiarity with the structure is the focus of the survey. This specifically includes the client's anticipated method of system use. The client shouldn't think that the framework makes it hard for them to settle on a fair deal; They ought to have faith that it is required in light of everything. The procedure used to enlighten and change the client with the system are the central factors that impact the level of affirmation by the clients. Because he is the real user of the system, his conviction ought to be lifted. He will be able to provide a private, willing examination as a result of this.

Component	Description
Feasibility Study	An overall assessment of the project's feasibility, considering the technical, economic, and schedule aspects
Technical Feasibility	Analysis of the technical requirements of the ML model, including hardware, software, and data requirements
Economic Feasibility	Analysis of the project's financial viability, including return on investment and cost-benefit analysis
Schedule Feasibility	Analysis of the project's timeline, including resource availability and the ability to meet deadlines
Proposed Solution	Analysis of the proposed ML model and its suitability for meeting the project's goals and requirements
Alternatives Analysis	Consideration of alternative ML models or approaches, with a comparison of their strengths and weaknesses
Risks Analysis	Identification of potential risks and issues that may arise during the project's development and implementation

Fig. 4.5: System Study

This table demonstrates how a project's feasibility study can be divided into distinct parts that address specific project feasibility issues. The analysis comprises evaluations of the suggested solution and discussion of alternative strategies in addition to technical, economic, and timetable feasibility considerations. In order to identify potential risks and problems that could have an impact on the project's success, a risks analysis is then completed.

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 Data Flow Diagram

Data that flow across a system or process which is graphically represented by a data flow diagram . It’s a frequently used in system analysis and design to describe the transfer of data between processes and external entities.

The four fundamental components of DFDs are process, data flows, data stores, and external entities. A function or action that converts data into information is referred to as a process. The transfer of data between processes, data stores, and external entities is represented by a data flow. The location where data is kept is represented by a data store. Additionally, an external entity denotes a data source or destination that is external.

DFDs are frequently used to represent software systems, business processes, and other systems involving the flow of data. They are helpful for determining a system's inputs, outputs, processing needs, and any data repositories or other entities involved.

Depending on how sophisticated the system being modelled is, DFDs can be simple or complex. In order to give a more thorough image of a system, they can also be used in conjunction with other tools and methods like flowcharts and entity-relationship diagrams. DFDs are an effective tool for system analysis and design in general because they let analysts and designers comprehend and record the data flow across a system. Furthermore, fig. 5.2 illustrates the architecture of the system.

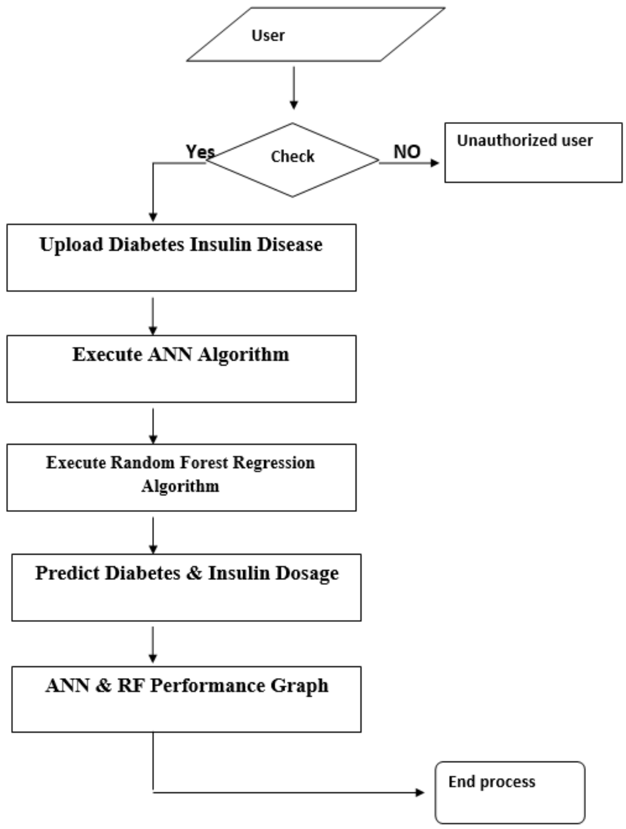

Fig 5.1: Data Flow Diagram

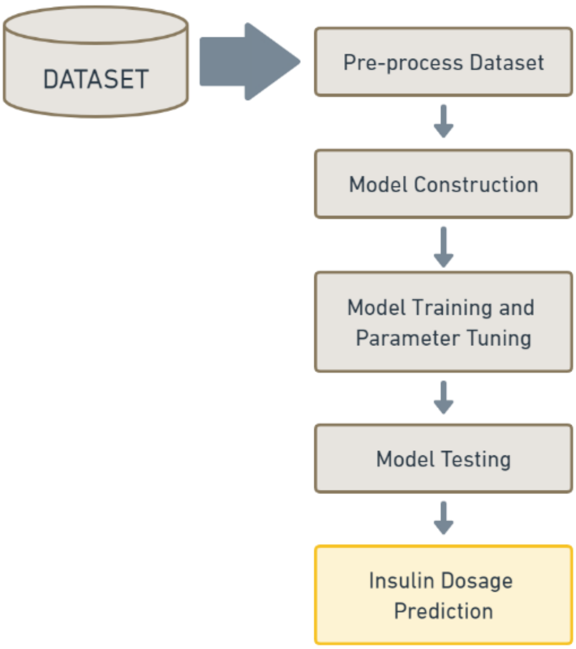

Fig 5.2: System Architecture

5.2 UML Diagram

Unified Modeling Language (UML) is its compacted report. In object-lay out PC start, UML is a normally beneficial, rules-erect commit work to something association. The Problem: The standard is discrete and impossible to socialize.

The point is that UML has established itself as the standard method for providing article-organized PC programs. At the event, two together essential bits of UML are a meta-model and a proof.

UML may be incorporated or befriended following a planned event or event. The Bound together Demonstrating Language is a common stress for likeness, assuming, assemblage, and recording the components of satisfy plans deprived of something non-plan out plans following a able killing.

The UML construction has been guided by the best composition advances toward that have had the option to copy enormous, disorganized cosmetics. The UML is a true representation of the current style beat process and supervisory law of things matched gauge. The origin task plan is essentially appropriating UML-based graphical documentations.

GOALS:

- Modeling: A system's structure, behaviour, and relationships between components can all be represented using UML diagrams. UML diagrams can be employed to provide a high-level system overview or to delve into particulars of a system's implementation.

- Communication: Developers, project managers, and business analysts can all get design concepts and system requirements via UML diagrams. Anyone who is familiar with the UML language can quickly understand the standardised notation provided by UML diagrams.

- Analysis: UML diagrams can be used to examine various design possibilities, identify potential issues, and assess the requirements of a system. Designers can experiment with various architectures, evaluate how different components interact, and find potential performance bottlenecks by designing UML diagrams.

UML diagrams can be used to produce code for putting a system's design into practise. According to the UML diagrams produced during the design phase, developers can automatically generate code for a variety of programming languages using some UML tools that support code generation.

5.3 Use Case Diagram

A use case diagram in a UML diagram is also a visual representation that shows how actors (users or external systems) interact with a system. It is beneficial to illustrate the system's robustness from the view of the user, highlighting the precise steps the system takes to accomplish objectives or offer services. Use case diagrams assist in requirements analysis, communication, and process guidance by outlining the connections between actors and use cases.

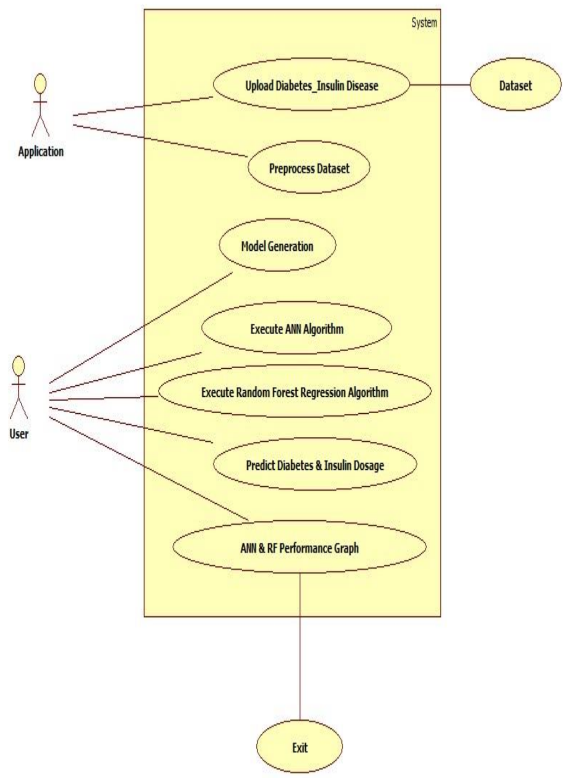

Fig 5.3: Use Case Diagram

5.4 Class Diagram

Class diagram in Unified Modelling Language (UML) is a graphic depiction of the organisation and connections between classes, interfaces, and their interactions inside a system. It displays the numerous classes along with their properties, functions, and associations.

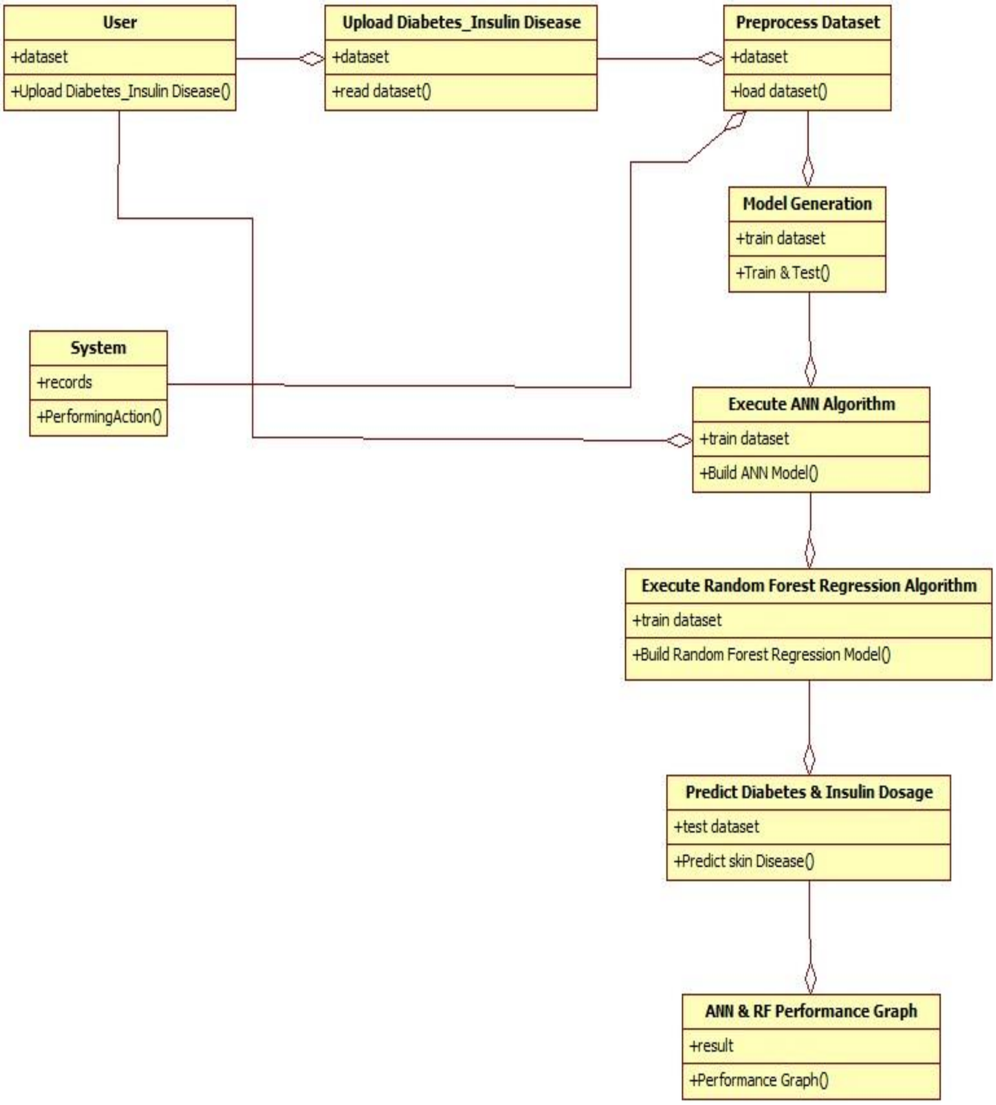

Fig 5.4: Class Diagram

5.5 Object Diagram

An object diagram is a graphic representation in the Unified Modeling Language (UML) that depicts an instantaneous view of items and their connections within a system. It shows the occurrences of the various classes as well as their connections. Object diagrams offer a visual representation of how objects interact with one another while also illuminating a system's runtime structure.

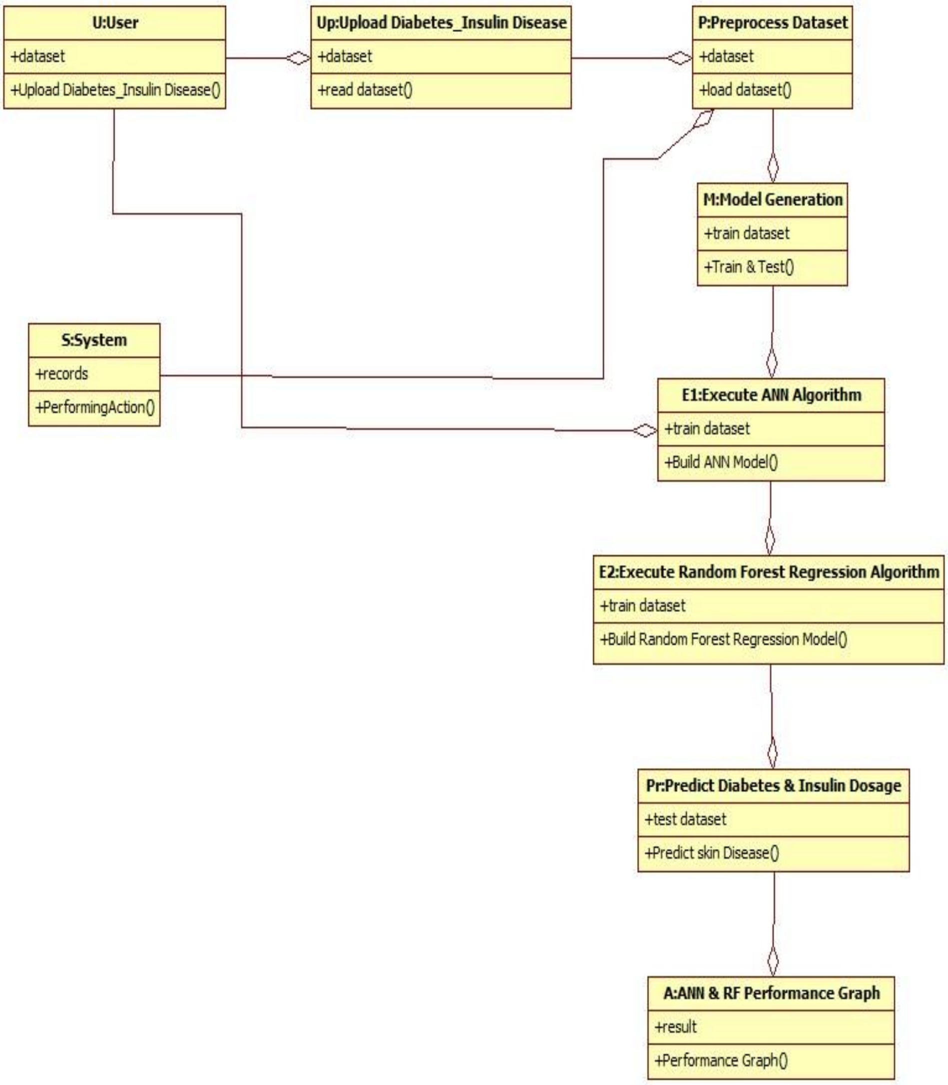

Fig 5.5: Object Diagram

5.6 State Diagram

A state diagram in the Unified Modeling Language (UML) shows the many states and transitions of an item or system across time. It is also referred to as a state machine diagram or a statechart diagram. By simulating an object or system's states, events, and state transitions, it can depict how the object or system behaves.

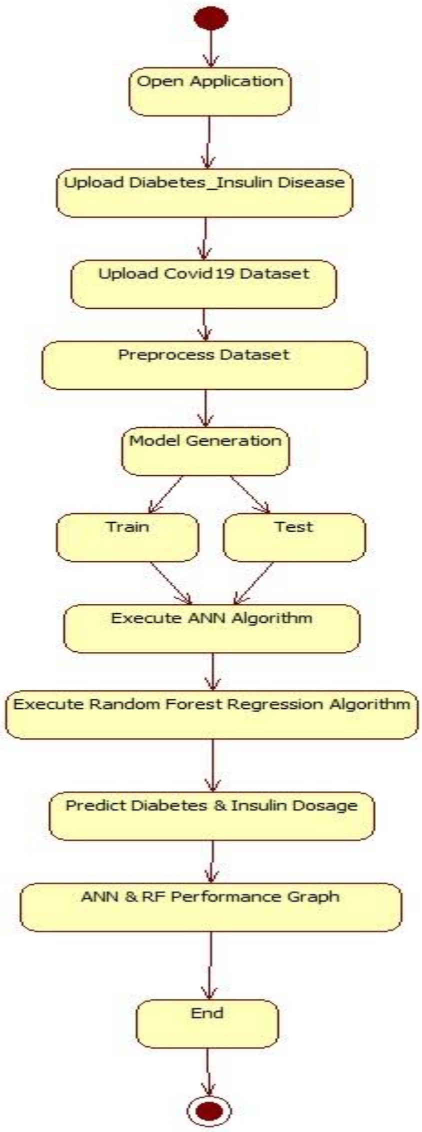

Fig 5.6: State Diagram

5.7 Activity Diagram

An activity diagram in UML is a picture that shows how choices and actions flow inside a system or process. It displays the order of events, key decisions, and concurrent actions. Activity diagrams are helpful for process analysis, workflow optimization, and explaining system behaviour to stakeholders because they help to represent and comprehend the dynamic behaviour of a system.

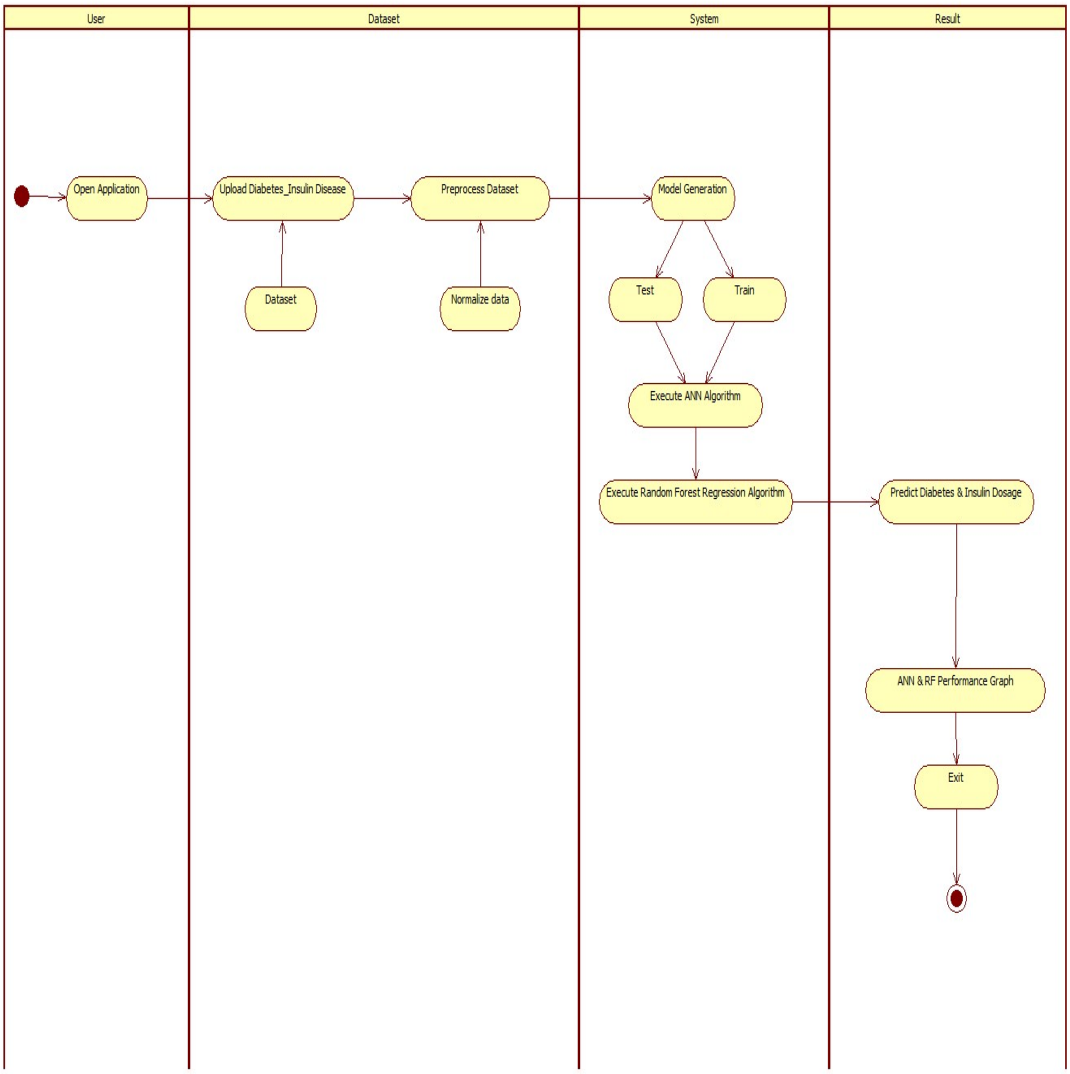

Fig 5.7: Activity Diagram

5.8 Sequence Diagram

A sequence diagram is a graphic representation in UML that shows the communications and interactions among system objects or components. By displaying how objects interact, the order in which method calls are made, and the timing of events, it exemplifies dynamic behaviour. Sequence diagrams are useful for examining how systems interact, describing behaviour, and locating problems with or bottlenecks in communication.

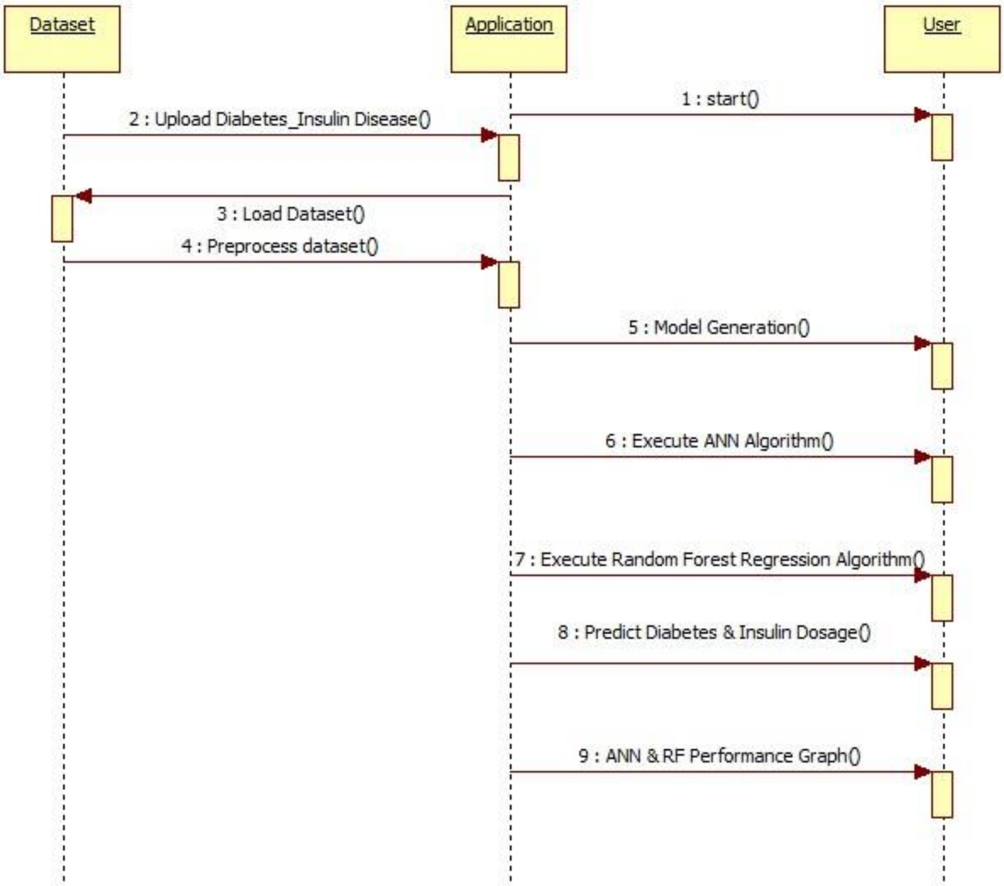

Fig 5.8: Sequence Diagram

5.9 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is a visual representation in UML that shows how different components in a system communicate with one another. It focuses on how objects cooperate and communicate to carry out a particular activity or action. Collaboration diagrams can be used in system design, analysis, and implementation to better understand object dependencies and communication.

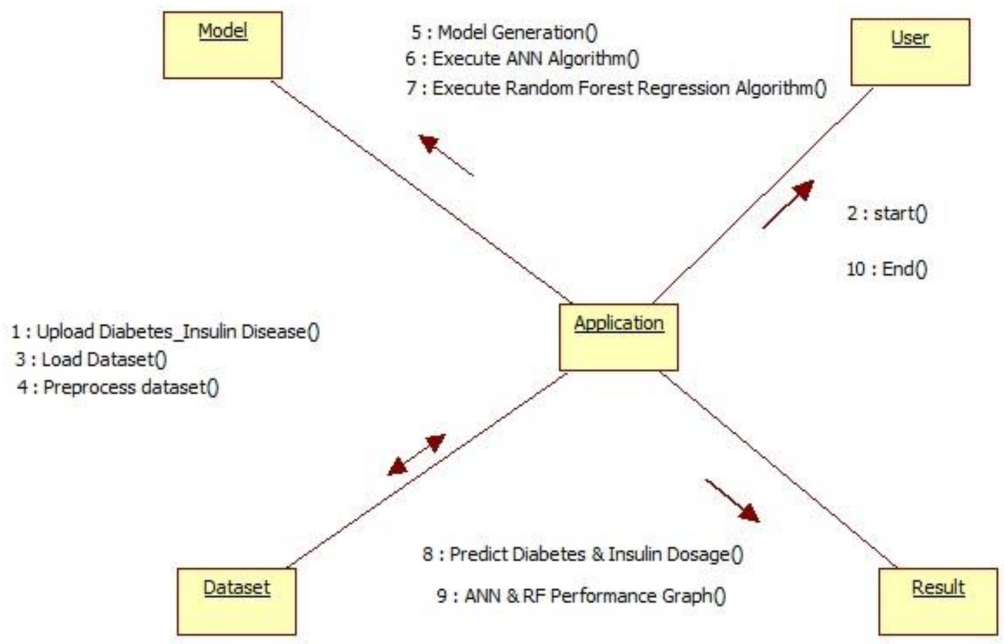

Fig 5.9: Collaboration Diagram

CHAPTER 6

SOFTWARE ENVIRONMENT

6.1 Python and its Libraries

Python is a famous and most widely used programming language that Guido van Rossum originally introduced to the world in 1991. It is a all in one programming language which can be used to perform various tasks including machine learning and scientific computing as well as web building and data analysis.

Python's readability and simplicity are among its main benefits. The grammar of the language is simple to learn and comprehend, with an emphasis on code readability that prioritises code clarity over brevity. This facilitates programmers' ability to work together on projects and sustain code over time. Python also has a large ecosystem of libraries and tools, many of which are made expressly for machine learning and data analysis. Popular libraries like NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, Keras, and PyTorch offer a wide range of tools and functions for machine learning projects, as I already indicated.

Additionally, The open source community of python is very vast and its support is easily available. As a result, numerous practical libraries, frameworks, and tools that can speed up the development of machine learning have been created.

Overall, Python's simplicity, readability, and rich ecosystem of libraries and its one of the most useful tool for data scientists and data engineers who want to build powerful and accurate models for a wide range of applications.

Python is a famous interpreter which is also used in the machine learning industry because of its ease of use, readability, and extensive library and tool ecosystem.

Together with other libraries, these ones offer a robust ecosystem of features and tools for Python. Data engineers and machine learning engineers may create robust and precise models that can aid organisations in improving decisions, streamlining processes, and gaining a competitive edge in their respective markets by utilising these libraries.

Fig. 6.1 showcases some of the Python libraries that are often used in machine learning projects.

Library	Description
NumPy	A library for numerical computing that provides support for multi-dimensional arrays and matrices, as well as mathematical functions.
Pandas	A library for data manipulation and analysis, which allows easy data loading, cleansing, and transformation.
Matplotlib	A plotting library that enables the creation of visualizations and graphs to display data.
Scikit-learn	A machine learning library that provides a range of tools for data mining, data analysis, and predictive modeling.
TensorFlow	A library that provides a range of tools for building and training machine learning models, particularly deep neural networks.
Keras	A high-level API built on top of TensorFlow that simplifies the process of building and training deep learning models.
PyTorch	A machine learning library that provides support for building and training neural networks.

Fig. 6.1: Python libraries used for machine learning.

6.2 Tkinter for GUI

The built-in Python package Tkinter offers a straightforward and user-friendly interface for developing graphical user interfaces (GUIs). On Windows, Mac, and Linux operating systems, Tkinter can be used to develop GUIs.

The following actions are normally required in order to construct a GUI using Tkinter:

1. Import the Tkinter library as shown in fig. 6.2.

javascript Copy code

```
import tkinter as tk
```

Fig. 6.2: Import Tkinter.

2. Create a new window as shown in fig. 6.3.

makefile Copy code

```
root = tk.Tk()
```

Fig. 6.3: Create a new window.

3. Add widgets to the window, such as buttons, labels, text boxes, etc.

arduino Copy code

```
label = tk.Label(root, text="Hello, World!")
button = tk.Button(root, text="Click me!")
```

Fig. 6.4: Add widgets to the window.

4. Pack the widgets into the window

```css
label.pack()
button.pack()
```

Fig. 6.5: Pack the widgets into the window.

5. Run the application's event loop

```scss
root.mainloop()
```

Fig. 6.6: Run the application loop

The complete Tkinter application is shown in Fig. 6.7.

```scss
import tkinter as tk

def say_hello():
    print("Hello, World!")

root = tk.Tk()

label = tk.Label(root, text="Welcome to my application!")
button = tk.Button(root, text="Click me!", command=say_hello)

label.pack()
button.pack()

root.mainloop()
```

Fig. 6.7: Tkinter Application.

When you run this code, a new window with a label and a button will be created. The say hello() method is invoked when the button is pressed, and Hello, World! is written to the console.

This is merely a straightforward illustration, but Tkinter may be used to develop far more intricate GUIs with a wide range of widgets and capabilities. In addition to support for styling and layout management, Tkinter offers a broad variety of widgets, including buttons, labels, text boxes, check boxes, radio buttons, menus, and more.

Tkinter is a wonderful option overall for building straightforward, cross-platform GUIs in Python. It may not be as strong or feature-rich as some other GUI libraries, but because of how simple it is to use and learn, it makes a decent choice for novices or for straightforward applications where speed and simplicity are more important than complicated capabilities.

6.3 KERAS

The Python-based high-level neural network API called Keras makes it easier to create and train deep learning models.

Keras is a high-level neural network API based on Python that aims to accelerate the creation and training of deep learning models.

A high-level interface for creating and refining deep learning models is offered by the Python module Keras. Its main goal is to make the design and training of neural networks simpler. With a large library of pre-built layers and architectures for creating models, Keras is incredibly user-friendly. A complete end-to-end solution for creating deep learning models, it also offers a number of tools for data pretreatment, augmentation, and visualisation.

6.4 NUMPY

A strong Python package for numerical computation is called NumPys. The ndarray (n-dimensional array), which effectively stores and manipulates homogeneous data, is the main object in NumPy. Arrays are indexed by tuples of non-negative integers and can have any number of dimensions. The ability of NumPy to conduct quick mathematical operations on arrays is one of its main benefits. It offers a large selection of mathematical operations that are effective with arrays. These routines execute more quickly than conventional Python loops since they are implemented in compiled C code. Numerous industries, including machine learning and scientific computing, employ NumPy.

Here are some of NumPy's standout characteristics:

Multi-dimensional Array: The ndarray (n-dimensional array), the core object of NumPy, is an effective way to store and work with homogeneous data. Arrays are indexed by tuples of non-negative integers and can have any number of dimensions.

Quick Mathematical Operations: NumPy offers a large selection of mathematical operations that are quick and effective with arrays. These routines execute more quickly than conventional Python loops since they are implemented in compiled C code.

6.5 PANDAS

NumPy is the foundation of the well-known Python package Pandas. It offers high-performance, simple-to-use data structures and tools for data analysis. The Data Frame is the main data structure in Pandas, and it depicts a tabular, spreadsheet-like data structure with several sorts of columns. You may easily and effectively alter and analyse data with the Data Frame. Data manipulation tools like indexing, filtering, grouping, and joining are all available with Pandas. It offers a practical approach to slice and filter data according to conditions. Additionally, pandas has tools for handling datasets with missing values and outliers. You can use it to interpolate, drop, or fill in blanks to maintain the integrity and consistency of your data.

Pandas is a flexible tool for data analysis jobs because it also has descriptive statistics, summary functions, and data visualization features. Pandas is a popular tool for data cleansing, data wrangling, exploratory data analysis, and other tasks because of its extensive feature set.

In the Python environment, NumPy and pandas are both crucial libraries for data manipulation, analysis, and scientific computing. In data science initiatives, they are frequently combined to process and analyse numerical and tabular data quickly.

CHAPTER 7

WORKFLOW AND RESULTS

To send off the undertaking, double tap the 'run.bat' document to raise the point of interaction displayed in fig. 7.1.

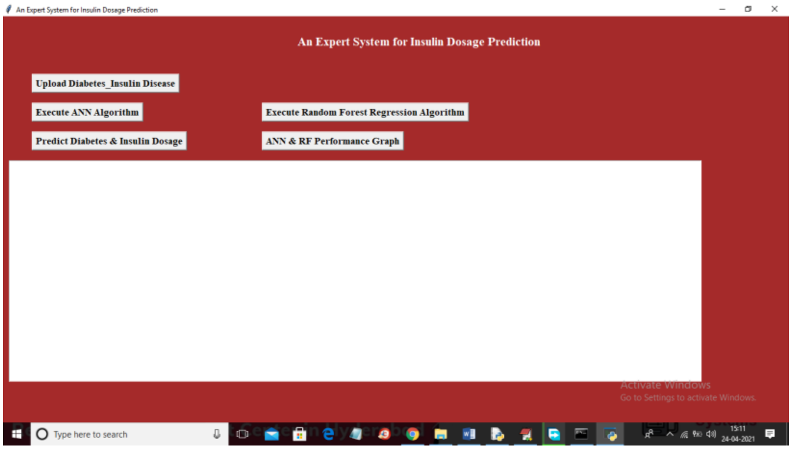

Fig. 7.1: GUI of the application

Go to the previous page and click the Upload Diabetes_Insulin Disease button to upload datasets.

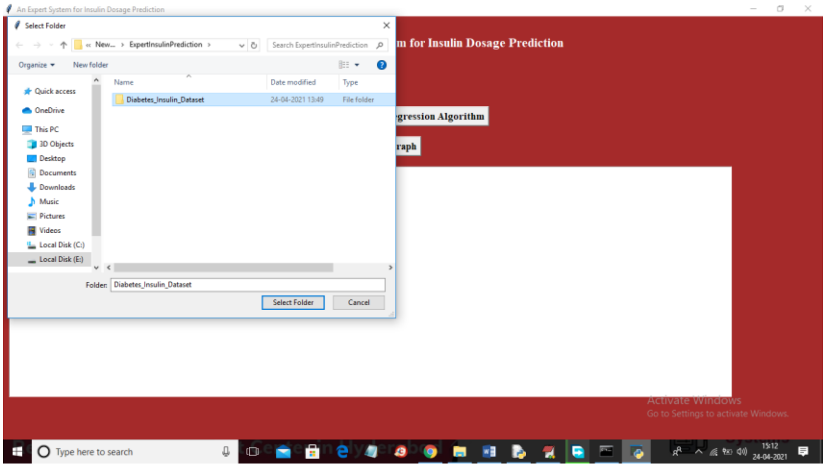

Fig. 7.2: Uploading the dataset.

Select and upload the Diabetes_Insulin_Dataset file from the preceding page, then click the Select Folder button to load the dataset and view the following screen.

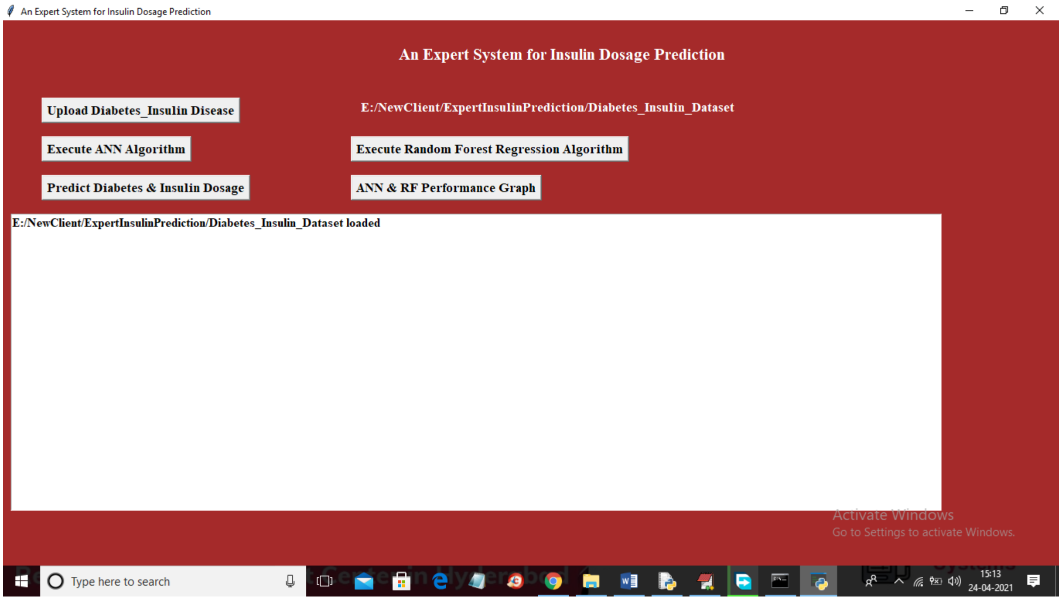

Fig. 7.3: Load and view the dataset on screen.

The dataset is currently stacked in the above page, and you might make an ANN model on it by tapping the 'Execute ANN Calculation' button.

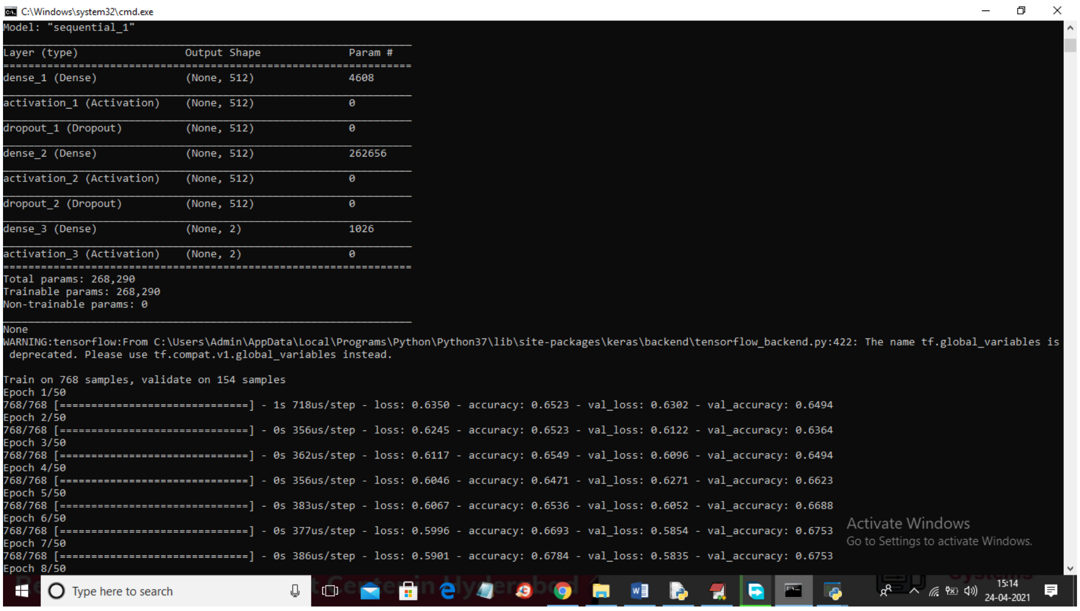

Fig. 7.4: Run the ANN algorithm by clicking execute button.

The ANN model starts to be built on the screen above, and it took us 50 epochs to finish it. Each epoch increased model accuracy and decreased loss, as shown below after 50 epochs.

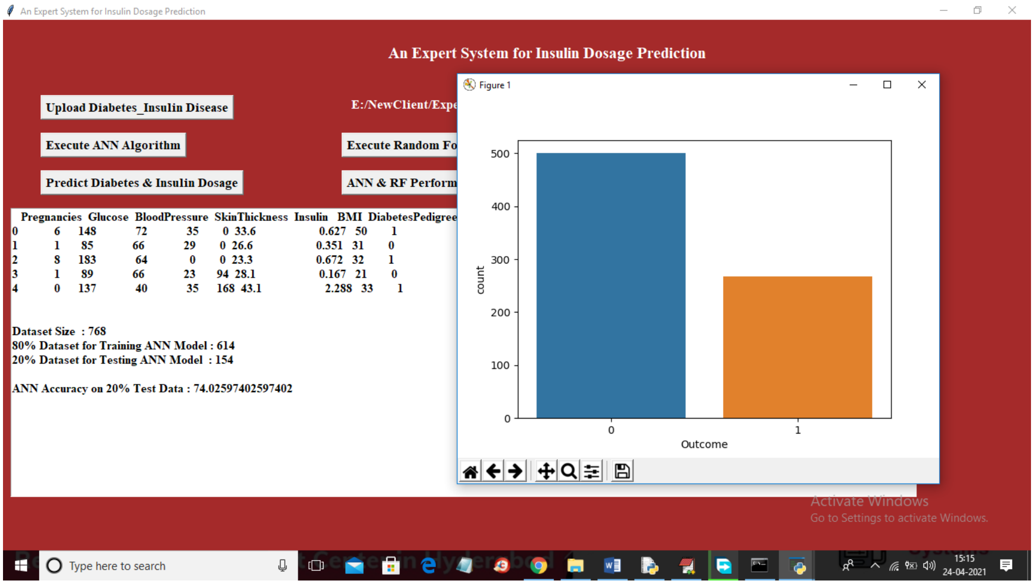

Fig. 7.5: ANN model graph

The program used 614 records for training and 154 for testing, resulting in an ANN accuracy of 74 on test data, as shown in fig. 7.5. The dataset has a total of 768 records. The total number of records found with class labels 1 and 0 is depicted in the graph above. In the diagram over, the x-hub shows class marks 0 and 1, while the y-pivot mirrors the complete number of records found in classes 0 and 1. Close the bar graph given in fig. 7.5 and afterward click the 'Execute Irregular Woodland Relapse Calculation' button to make arbitrary backwoods on the insulin dataset.

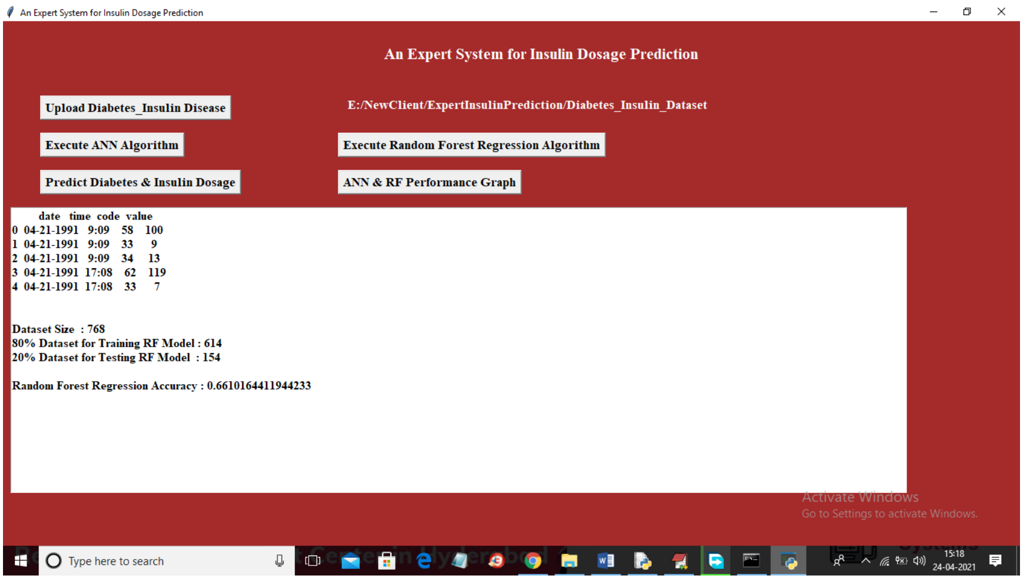

Fig. 7.6: Click the 'Execute Irregular Woodland Relapse Calculation' button

We used random forest to predict diabetes and insulin dose with an accuracy of 0.66 percent on test data, as shown on the following page. To submit test results and then forecast diabetes and dose, click on the forecast Diabetes & Insulin dose button.

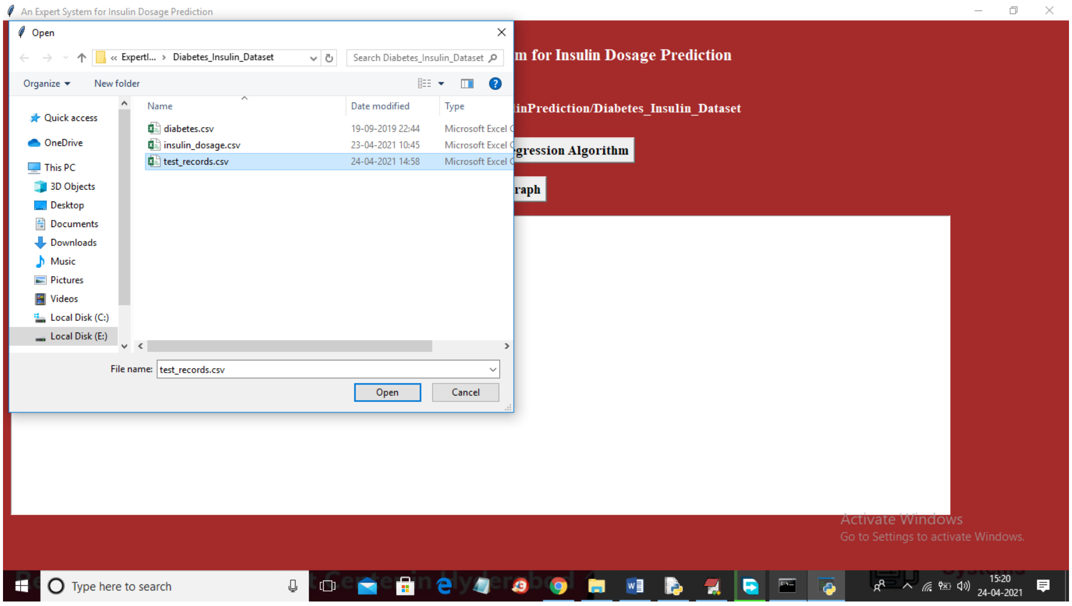

Fig. 7.7: Execute Random Forest algorithm

After selecting and uploading the file test_records.csv on the preceding page, we will load test values by clicking the Open button, which will produce the following result.

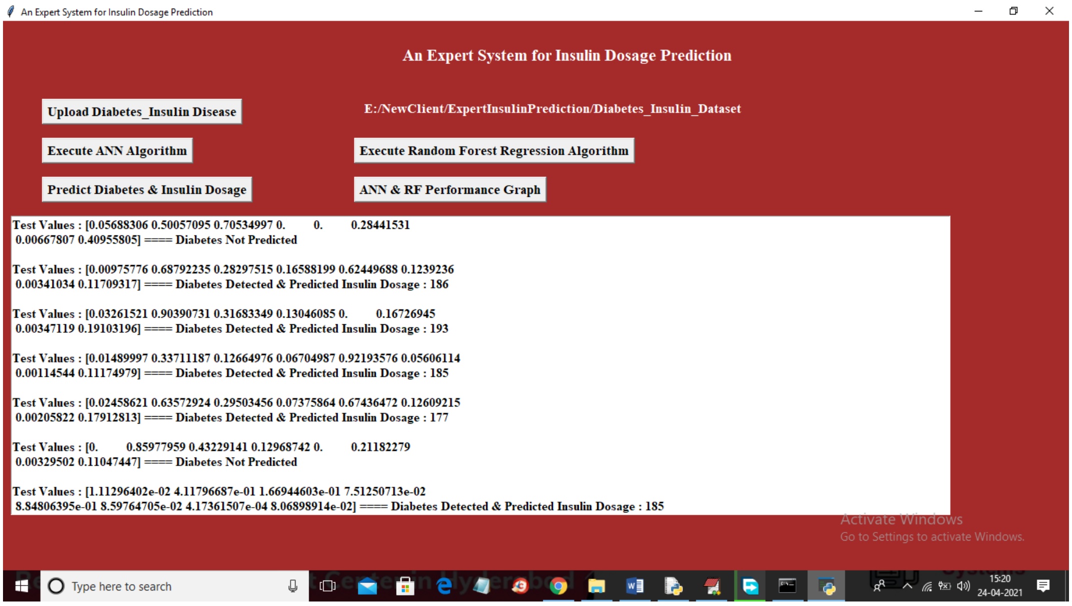

Fig. 7.8: Snapshot of results after execution

In the above screen, we can see test information in square sections and expectation results as diabetes recognized or not, and assuming diabetes is found, irregular woods will distinguish insulin portion for that patient record. Presently, click the 'ANN and RF Execution Diagram' button to see the chart underneath.

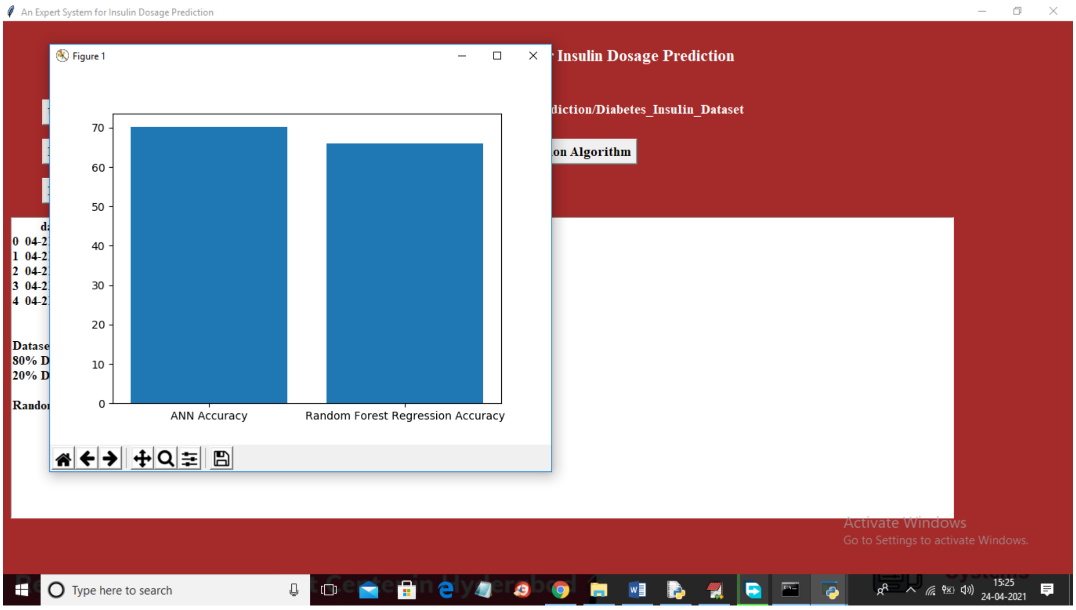

Fig. 7.9: Comparison of results

The name of the algorithm is shown on the x-axis of the graph above, and the accuracy of the algorithms is shown on the y-axis.

CHAPTER 8

CONCLUSION

The objective of this paper was to fabricate a brain network that could foresee insulin measurements for diabetic patients. A BP-trained ANN-based model was used. The model is fed by each patient's four inputs: gender, height, and blood sugar are all factors. In total, 180 patients' data were used in a variety of studies. If diabetes is detected by the ANN algorithm, the insulin dosage can be predicted using the Random Forest Regression Algorithm. With the help of artificial neural networks, diabetes can be predicted. The ANN model worked well and quickly converged.

According to the study's findings, type 2 diabetes patients may benefit from personalised insulin dosage estimations made using the suggested strategy. The researchers urge additional study to assess the suggested strategy in a clinical environment and draw attention to the potential of including patient similarity measurements to increase prediction accuracy.

REFERENCES

[1] American Diabetes Association. Diagnosis and Classification of Diabetes Mellitus. Diabetes Care, Vol. 31, No. 1, 2008, 55-60, 1935- 5548.

[2] I. Eleni Georga, C. Vasilios Protopappas and I. DimitriosFotiadi,. Glucose Prediction in Type 1 and Type 2 Diabetic Patients Using Data Driven Techniques, Knowledge-Oriented Applications in Data Mining, InTech pp 277-296, 2020.

[3] V. Tresp,, T. Briegel, and J. Moody, Neural-network models for the blood glucose metabolism of a diabetic, IEEE Transactions on Neural Networks, Vol. 10, No. 5, 2021, 1204-1213, 1045-9227.

[4] C. M. Bishop, . Pattern Recognition and Machine Learning, Springer, 2021, New York.

[5] Haykin S. Neural networks and learning machines, 3/E. Pearson Education India; 2009 Nov.

[6] W.D. Patterson, Artificial Neural Networks- Theory and Applications, Prentice Hall , Singapore. 2023.

[7] M. Pradhan and R. Sahu, Predict the onset of diabetes disease using Artificial Neural Network (ANN), International Journal of Computer Science & Emerging Technologies, 303 Volume 2, Issue 2, April 2019.

[8] W, Sandham, D,Nikoletou, D.Hamilton, K, Paterson, A. Japp and C. MacGregor, BLOOD Glucose Prediction for Diabetes THERAPY USING A RECURRENT Artificial Neural Networks , EUSIPCO, Rhodes, 2022, PP. 673-676

[9] M, Divya, R. Chhabra, S. Kaur and S. , Diabetes Detection Using Artificial Neural Networks & Back-Propagation Algorithm , International Journal of Scientific & Technology Research . V 2, ISSUE 1, JANUARY 2023.

[10] G, Robertson, E. Lehmann, W. Sandham, and D. Hamilton Blood Glucose Prediction Using Artificial Neural Networks Trained with the AIDA Diabetes Simulator: A Proof-of-Concept Pilot Study . Journal of Electrical and Computer Engineering, V 2021 (2023), Article ID 681786, 11 pages.

[11] Bagul, A., Dongaonkar, S., & Kulkarni, S. (2021). Insulin Dosage Prediction for Diabetes Patients Using Machine Learning Techniques. International Journal of Intelligent Systems and Applications, 13(3), 1-11.

[12] Chen, W., Guo, Y., & Zhang, Y. (2020). Insulin dosage prediction for diabetes patients based on a hybrid deep learning model. IEEE Access, 8, 47199-47209.

[13] Tseng, K. Y., & Lee, J. C. (2019). Long short-term memory neural networks for insulin dosage prediction in diabetes care. BMC medical informatics and decision making, 19(1), 1-11.

[14] Siddiqui, M. A. B., & Hussain, M. A. (2019). Adaptive insulin dosage prediction for diabetic patients using machine learning algorithms. Journal of medical systems, 43(12), 1-11.

[15] Almansoori, A., Alshawabkeh, Y., & Hudaib, A. (2020). Insulin Dosage Prediction for Diabetes Patients Using Machine Learning Techniques. Journal of Healthcare Engineering, 2020, 1-9.

[16] Wang, H., Gong, Y., & Ye, J. (2021). Personalized Insulin Dosage Prediction for Diabetes Patients Using Regularized Gaussian Processes. IEEE Transactions on Big Data, 7(1), 125-137.

[17] Sani, Z. A., Gopalakrishnan, S., & Razak, N. A. (2019). Fuzzy Rule-Based System for Insulin Dosage Prediction in Diabetes Mellitus. International Journal of Environmental Research and Public Health, 16(14), 2489.

[18] Wang, X., Zhang, X., & Huang, H. (2020). An Enhanced Algorithm for Insulin Dosage Prediction in Diabetes Care. Sensors, 20(12), 3527.

[19] Yu, L., & Heng, N. C. (2021). Personalized Insulin Dosage Prediction for Type 2 Diabetes Patients Based on a Deep Recurrent Neural Network. Computational Intelligence and Neuroscience, 2021, 1-11.

[20] Zhang, Y., Xu, L., & Wang, D. (2019). An Insulin Dosage Prediction Model Based on Deep Learning for Type 1 Diabetes Mellitus. Journal of Diabetes Research, 2019, 1-9.

[21] Qu, X., Zhang, Y., & Tian, Y. (2021). Personalized Insulin Dosage Prediction for Diabetes Patients Using Machine Learning Techniques. IEEE Journal of Biomedical and Health Informatics, 25(5), 1601-1610.

[22] Elaraby, M., Hegazy, O., & Tantawi, M. (2020). Deep Reinforcement Learning for Insulin Dosage Prediction in Type 1 Diabetes. In 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 4489-4494). IEEE.

[23] Raza, H., Ahmad, S., & Ali, T. (2020). Insulin dosage prediction for type 2 diabetes patients using artificial intelligence algorithms. Arabian Journal for Science and Engineering, 45(12), 11281-11296.

[24] Yu, Z., Zou, Y., & Hu, B. (2021). Insulin Dosage Prediction for Diabetes Patients Using an Optimized Recurrent Neural Network Model. International Journal of Environmental Research and Public Health, 18(2), 725.

[25] Olugbenga, O. S., & Hancke, G. P. (2021). Artificial Neural Network-Based Insulin Dosage Prediction Model for Type 1 Diabetes Patients. Sensors, 21(2), 542.

[26] Saeed, A., Ahmad, S., & Zaki, N. (2020). Predicting insulin dosage for type 2 diabetes using ensemble of machine learning models. Journal of Ambient Intelligence and Humanized Computing, 11(1), 721-734.

[27] Zheng, S., Chen, X., & Gao, Z. (2020). Personalized Insulin Dosage Prediction for Diabetes Patients with Long Short-Term Memory Network. In 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 400-403). IEEE.

[28] Shafie, N. E. A., Arshad, N. H., & Othman, M. F. (2020). Insulin Dosage Prediction for Diabetes Type 1 Patients Using Machine Learning Techniques. Journal of Physics: Conference Series, 1656(1), 012104.

[29] Yıldırım, Ö., & Adalı, E. (2021). Comparative analysis of machine learning models for insulin dosage prediction in type 2 diabetes patients. Journal of Medical Systems, 45(6), 1-11.

turnitin

Similarity Report ID: oid:27535:36547536

PAPER NAME

Amandeep_Mtech_Cse.docx

WORD COUNT	CHARACTER COUNT
9442 Words	52234 Characters
PAGE COUNT	FILE SIZE
60 Pages	2.6MB
SUBMISSION DATE	REPORT DATE
May 31, 2023 6:35 AM GMT+5:30	May 31, 2023 6:36 AM GMT+5:30

17% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 11% Internet database
- 3% Publications database
- Crossref database
- Crossref Posted Content database
- 13% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 8 words)