Diabetes Prediction using AI and Machine Learning

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with
TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Sri Parvathi Devi Kurakula, ksparvathid@gmail.com

Under the Guidance of

Sowmya Chowdary

ACKNOWLEDGEMENT

We would like to take this opportunity to express our deep sense of gratitude to all individuals who helped us directly or indirectly during this thesis work.

The successful completion of any task is not possible without proper Suggestions, guidance, and environment. The combination of these three Factors acts as a backbone to my "Diabetes Prediction using Fused Machine Learning" project.

Firstly, we would like to thank my supervisor, Sowmya Chowdary, for being a great mentor and the best adviser I could ever have. His advice, encouragement and the critics are a source of innovative ideas, inspiration and causes behind the successful completion of this project. The confidence shown in me by him was the biggest source of inspiration for me. It has been a privilege working with him for the last one year. He always helped me during my project and many other aspects related to the program. His talks and lessons not only help in project work and other activities of the program but also make me a good and responsible professional.

ABSTRACT

Diabetes is a most common disease caused by a group of metabolic disorders. It is also known as Diabetic mellitus. In the medical field, it is essential to predict diseases early to prevent them. In modern lifestyles, sugar and fat are typically present in our dietary habits, which have increased the risk of diabetes To predict the disease, it is extremely important to understand its symptoms . Early prediction in such disease can be controlled and save human life.

Although the research on diabetes prediction has made great progress,prediction with high accuracy is still a big challenge.Currently, machine-learning (ML) algorithms are valuable for disease detection.

The proposed system has used fused machine learning approach for diabetes prediction.. The dataset used in this research is divided into training data and testing data respectively. The output of these models becomes the input for voting classifier whereas the voting classifer combines the different algorithms into one and determines whether a diabetes diagnosis is positive or negative.. we stores the fused models for future use on internally. Based on the patient's real-time input, the fused model predicts whether the patient is diabetic or not.In this paper,Artificial Neural Networks(ANN) and Random Forest algorithms are used .

A fused machine learning approach combines multiple machine learning models to enhance predictive performance and accuracy. This method leverages the strengths of different algorithms, thereby mitigating the weaknesses of individual models.

Keywords: Fused machine learning, Diabetes prediction ,voting classifier, ANN algorithm, .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

1. ML	Machine Learning
2. SVM	Support Vector Machine
3. ANN	Artifical Nueral Network
4. MLP	Multilayer Perceptron
5. HTML	Hyper Text Markup Language
6.RF	Random Forest
7. OS	Operating System
8. UML	Unified Modelling Language
9. URL	Uniform Resource Locator

CHAPTER-1

INTRODUCTION

CHAPTER-1

INTRODUCTION

1.1 Diabetes

Diabetes is a chronic medical condition that occurs when the body is unable to properly regulate blood sugar (glucose) levels. Glucose is a crucial source of energy for the cells in our body, and its levels are tightly controlled by the hormone insulin, which is produced by the pancreas. There are two main types of diabetes: Type 1 and Type 2.

Type 1 Diabetes: In Type 1 diabetes, the immune system mistakenly attacks and destroys the insulin-producing beta cells in the pancreas. As a result, the pancreas is unable to produce enough insulin, leading to a lack of insulin in the body.

People with Type 1 diabetes require insulin injections or an insulin pump to replace the insulin their bodies can no longer produce.

Type 2 Diabetes: Type 2 diabetes is characterized by insulin resistance, where the body's cells do not respond effectively to insulin. Initially, the pancreas compensates by producing more insulin to maintain normal blood sugar levels.

Over time, the pancreas may become unable to produce enough insulin, leading to elevated blood sugar levels.

Type 2 diabetes is often associated with lifestyle factors such as obesity, lack of physical activity, and genetics.

1.1.1 KEY ASPECTS OF DIABETES PREDICTION

Key aspects of diabetes prediction encompass various crucial elements .

1. Data Collection and Preparation:

Dataset Overview: The dataset includes demographic and clinical features: Age, Sex, Polyuria, Polydipsia, Sudden Weight Loss, Weakness, Polyphagia, Genital Thrush, Visual Blurring, Itching, Irritability, Delayed Healing, Partial Paresis, Muscle Stiffness, Alopecia, Obesity, and the target variable, Class (Positive/Negative).

Data Quality: Ensure data completeness, correctness, and consistency. Handle missing values and outliers.

2. Feature Engineering:

Binary Encoding: Convert categorical features (Yes/No) to binary values (0/1) for model compatibility.

Age Binning: Consider grouping age into bins (e.g., 20-30, 31-40) if it improves model performance.

Interaction Features: Explore interactions between features (e.g., Polyuria and Polydipsia together may indicate a higher risk).

3. Model Development:

ANN (Artificial Neural Network):

Architecture Design: Determine the number of layers, neurons per layer, activation functions, and dropout rates.

Training: Use backpropagation and optimization techniques (e.g., Adam optimizer) to minimize loss.

Random Forest:

Tree Construction: Choose the number of trees, depth of trees, and other hyperparameters.

Feature Importance:Utilize the inherent feature importance of RF to understand the contribution of each feature.

4. Model Fusion:

Combining Outputs: Use techniques such as:

Voting Classifier: A Voting Classifier is an ensemble learning technique that combines the predictions from multiple machine learning models to improve overall performance. In the context of diabetes prediction, using a Voting Classifier to fuse Artificial Neural Networks (ANN) and Random Forest (RF) models can leverage the strengths of both methods.

How Voting Classifier Works:

1. **Individual Models:** Train multiple base models (e.g., ANN and RF) on the same dataset.
2. **Voting Mechanism:** Combine the predictions from these models using either:
   - **Hard Voting:** Each model votes for a class, and the class with the majority votes is the final prediction.
   - **Soft Voting:** Each model provides a probability for each class, and the class with the highest average probability is the final prediction.

Advantages of Using Voting Classifier:

- **Improved Accuracy:** Combining models can lead to higher predictive accuracy than individual models.
- **Robustness:** Mitigates the weaknesses of individual models by leveraging their combined strengths.
- **Simplicity:** Easy to implement and interpret compared to more complex ensemble methods like stacking.

5. Evaluation Metrics:

Accuracy:Overall correctness of the model's predictions.

Precision: Ratio of true positive predictions to total positive predictions.

Recall (Sensitivity): Ratio of true positive predictions to actual positives.

F1-Score:Harmonic mean of precision and recall, balancing both metrics.

1.1.2 MACHINE LEARNING TECHNOLOGY

Machine learning is a branch of artificial intelligence that develops algorithms by learning the hidden patterns of the datasets used it to make predictions on new similar type data, without being explicitly programmed for each task.

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn from data and make predictions or decisions without explicit programming. The fundamental idea behind machine learning is to use data to identify patterns, make predictions, or optimize performance, allowing the system to improve its performance over time.

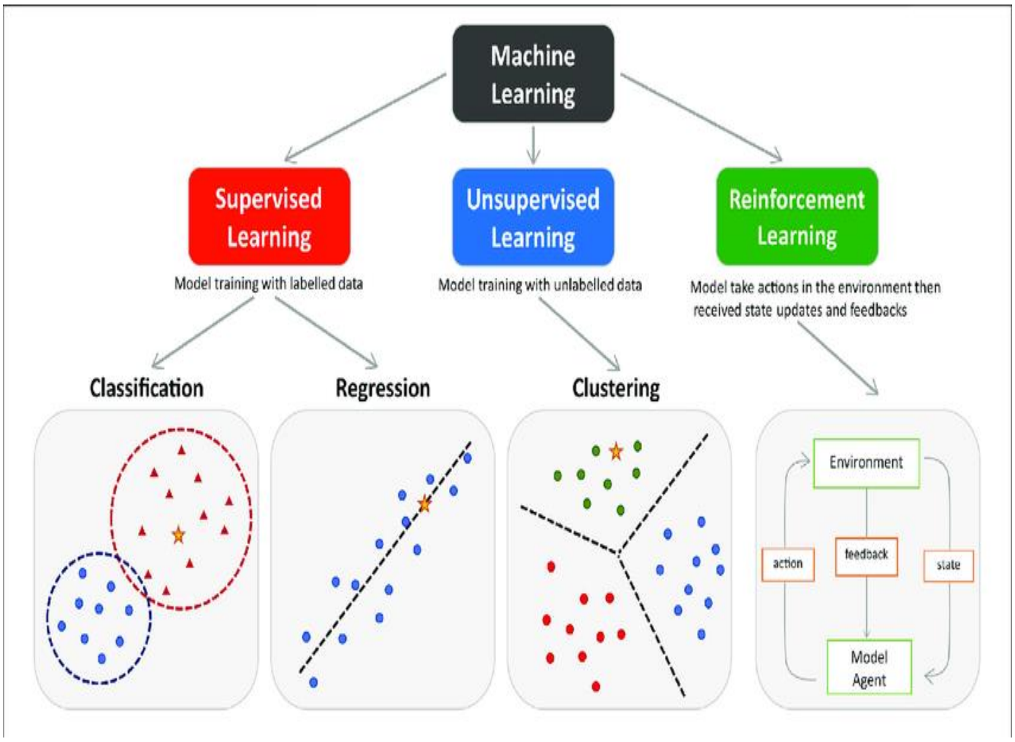


Fig 1.1.2 Machine learning Technology

Key Concepts in Machine Learning:

Data: Machine learning algorithms require data to learn from. This data can come in various forms, such as text, images, numerical values, or structured data from databases.

Algorithms: These are the mathematical models and procedures used by machine learning systems to learn patterns from data and make predictions or decisions. Common algorithms include decision trees, neural networks, support vector machines, and clustering algorithms.

Training: Machine learning models are trained using labeled or unlabeled data. In supervised learning, models are trained on labeled data where each input is associated with a known output. Unsupervised learning involves training on unlabeled data to identify patterns and structures.

Evaluation: After training, models are evaluated to assess their performance using metrics such as accuracy, precision, recall, or mean squared error. This step ensures that the model can generalize well to new, unseen data.

Deployment: Once trained and evaluated, models are deployed into production systems where they can make predictions or decisions on new data.

1.2 MACHINE LEARNING ARCHITECTURE

Machine learning architecture refers to the overall design and structure of a machine learning system. It encompasses the processes, components, and interactions required to develop, train, deploy, and maintain machine learning models. Here is a brief overview of the typical components and workflow in a machine learning architecture:

1. Data Collection

Data Sources: Raw data is collected from various sources such as databases, APIs, sensors, or web scraping.

Data Storage: The collected data is stored in databases, data lakes, or cloud storage solutions.

2. Data Processing

Data Cleaning: Removing noise, handling missing values, and correcting inconsistencies in the data.

Data Transformation: Converting data into a suitable format, which may include normalization, scaling, encoding categorical variables, and feature extraction.

3. Data Exploration and Visualization

Exploratory Data Analysis (EDA): Using statistical techniques and visualization tools to understand the data distribution, identify patterns, and detect anomalies.

Visualization Tools: Tools like matplotlib, seaborn, or Tableau are used to create visual representations of data.

4. Feature Engineering

Feature Selection: Identifying the most relevant features that contribute to the predictive power of the model.

Feature Creation: Generating new features from existing data to enhance model performance.

5. Model Selection

Algorithm Choice: Selecting the appropriate machine learning algorithm based on the problem type (classification, regression, clustering, etc.) and data characteristics.

Model Architecture: Designing the architecture of the model, particularly for complex models like neural networks.

6. Model Training

Training Data: Splitting the data into training and validation sets.

Training Process: Feeding the training data to the algorithm and adjusting the model parameters to minimize the error.

Hyperparameter Tuning: Optimizing the model by adjusting hyperparameters to improve performance.

7. Model Evaluation

Evaluation Metrics: Using metrics such as accuracy, precision, recall, F1-score, or mean squared error to evaluate the model's performance on the validation set.

Cross-Validation: Using techniques like k-fold cross-validation to assess model performance and avoid overfitting.

8. Model Deployment

Deployment Environment: Setting up the infrastructure to deploy the model, which can be on-premises servers, cloud platforms, or edge devices.

Model Serving: Exposing the model via APIs or integrating it into applications to make predictions on new data in real-time or batch mode.

9. Monitoring and Maintenance

Performance Monitoring: Continuously monitoring the model's performance in production to detect drift or degradation.

Model Updating: Periodically retraining the model with new data to maintain accuracy and relevance.

10. Security and Compliance

Data Privacy: Ensuring that data handling complies with regulations like GDPR or HIPAA.

Security: Implementing measures to secure data and models against unauthorized access or attacks.

1.2.1 TYPES OF MACHINE LEARNING

There are three main types of machine learning: supervised learning, unsupervised learning, and reinforcement learning.

1. Supervised Learning:

- In supervised learning, the algorithm is trained on a labeled dataset, where each input is associated with the corresponding output.

- The model learns the mapping between input features and the target output, making it capable of making predictions on new, unseen data.

- Common applications include image recognition, speech recognition, and regression tasks.

2. Unsupervised Learning:

- Unsupervised learning involves training the model on unlabeled data, and the algorithm tries to find patterns or structure within the data.

- It is used for tasks like clustering and dimensionality reduction.

- Clustering algorithms group similar data points together, while dimensionality reduction techniques aim to reduce the number of features in the dataset.

3. Reinforcement Learning:

- Reinforcement learning involves an agent learning to make decisions by interacting with an environment.

- The agent receives feedback in the form of rewards or punishments, allowing it to learn the optimal behavior over time.

- This type of learning is commonly used in robotics, gaming, and autonomous systems.

1.2.2 APPLICATIONS OF MACHINE LEARNING

Natural Language Processing (NLP): Understanding and generating human language, used in chatbots, sentiment analysis, and language translation.

Computer Vision: Interpreting visual data from the world, used in image recognition, object detection, and autonomous driving.

Healthcare: Predicting patient outcomes, diagnosing diseases from medical images, and personalized treatment recommendations.

Finance: Fraud detection, stock market prediction, and credit scoring.

Recommendation Systems: Personalizing recommendations for products, movies, or content based on user preferences.

1.2.3 Challenges in Machine Learning:

- **Data Quality**: Ensuring data is accurate, relevant, and representative of the problem domain.
- **Overfitting and Underfitting**: Balancing model complexity to avoid memorizing the training data (overfitting) or failing to capture important patterns (underfitting).
- **Interpretability**: Understanding how and why a model makes certain predictions, especially in critical applications like healthcare and finance.

Machine learning continues to advance rapidly, driven by increasing computational power, improved algorithms, and the availability of large datasets. It plays a crucial role in many industries, transforming how businesses operate and how researchers solve complex problems.

1.3 MACHINE LEARNING IN HEALTH CARE

Machine Learning in Healthcare:Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn from data and make predictions or decisions without explicit programming. The fundamental idea behind machine learning is to use data to identify patterns, make predictions, or optimize performance, allowing the system to improve its performance over time.

Benefits:

- Handles large datasets efficiently.
- Identifies complex patterns that are difficult to detect manually.
- Improves accuracy of predictions over traditional statistical methods.

1.3.1 Applications of Machine Learning in Healthcare:

Disease Prediction and Diagnosis:

Early Detection: ML algorithms can analyze patient data to predict the likelihood of developing conditions like diabetes, cancer, and cardiovascular diseases, enabling early intervention.

Predictive Analytics :

Outcome Prediction: ML models predict patient outcomes, such as the likelihood of hospital readmissions or complications, allowing healthcare providers to take preventive measures.

Risk Assessment: ML can assess patient risk factors for various conditions, helping in stratifying patients based on their health risks and providing targeted care.

1.4 ALGORITHMS

1.4.1Artificial Neural Networks

An artificial neural network (ANN) is a computational model inspired by the way biological neural networks in the human brain process information. ANNs are used in machine learning to recognize patterns, classify data, and make predictions. They consist of layers of interconnected nodes, or neurons, where each connection has a weight that is adjusted during training to optimize the network's performance.

Key Components of an Artificial Neural Network:

1. **Neurons (Nodes)**: Basic units that receive inputs, process them, and pass the outputs to the next layer. Each neuron applies a weighted sum of inputs followed by an activation function.
2. **Weights**: Parameters that determine the strength of the connection between neurons. These are adjusted during training to minimize the error in predictions.
3. **Bias**: An additional parameter added to the input of a neuron to allow the activation function to be shifted.
4. **Activation Function**: A non-linear function applied to the input of a neuron to introduce non-linearity into the network, allowing it to learn complex patterns. Common activation functions include sigmoid, tanh, and ReLU (Rectified Linear Unit).

Architecture of an Artificial Neural Network:

1. **Input Layer**: The first layer of the network that receives the raw input data.
2. **Hidden Layers**: Intermediate layers between the input and output layers. These layers perform complex transformations on the input data by passing it through neurons.
3. **Output Layer**: The final layer that produces the output of the network, which can be a classification, regression, or other types of prediction.

In this paper we used MLP neural network

Multilayer Perceptron (MLP)

- **Structure**: Consists of an input layer, one or more hidden layers, and an output layer.
- **Function**: Each neuron in a layer is fully connected to all neurons in the next layer.
- **Applications**: Used for tasks like classification, regression, and pattern recognition.

In summary, artificial neural networks are powerful tools for modeling complex patterns in data, and various types of ANNs are suited to different types of tasks. The MLP is one of the simplest forms, used for basic classification and regression problems

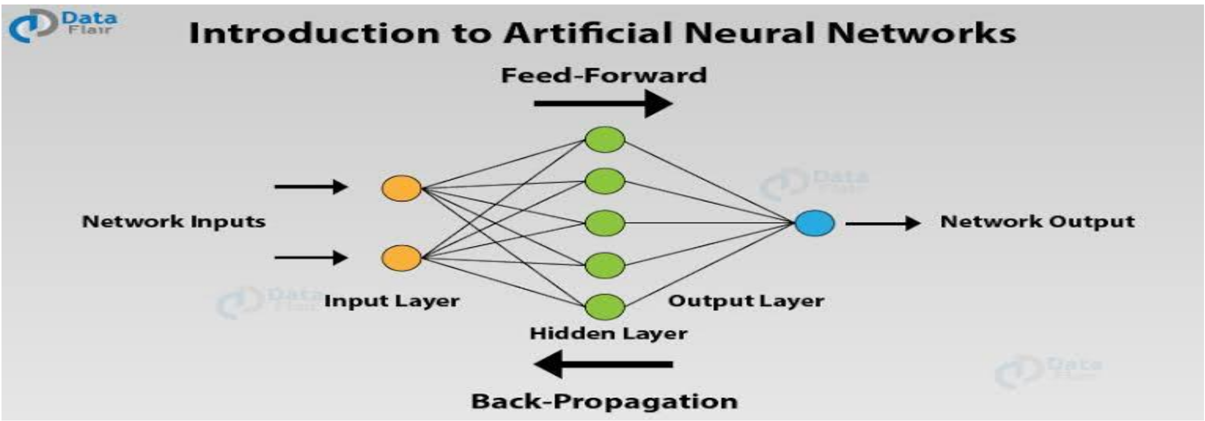


Fig 1.4.1 Introduction to Neural Networks

1.4.2 RANDOM FOREST

Random Forest is an ensemble learning method used for classification, regression, and other tasks that operate by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It was introduced by Leo Breiman and Adele Cutler and is known for its robustness and accuracy.

Key Concepts of Random Forest:

1. **Ensemble Learning**: The technique of combining multiple models to improve the overall performance. Random Forest is an example of ensemble learning, where multiple decision trees are combined to produce a better result than any single tree.
2. **Decision Trees**: The base models used in a Random Forest. Each decision tree is built using a random subset of the training data and features, and makes predictions based on the splits defined during its construction.
3. **Bagging (Bootstrap Aggregating)**: The method of creating multiple subsets of the original dataset by sampling with replacement. Each subset is used to train a different decision tree, which helps in reducing variance and avoiding overfitting.

1. 5 EXISTING SYSTEM

The existing fused decision model has achieved the accuracy of 95 %, which is higher than the other existing systems . Through this diagnosis model, we can save several lives. Moreover, the death ratio of diabetes can be controlled if the disease is diagnosed and preventative measures are taken in early-stage.The proposed system predicts the diabetes with higher accuracy.

DRAWBACKS OF EXISTING SYSTEM

- The models used in existing system often suffer from limitations, including lower accuracy and inadequate handling of data complexities. Preprocessing methods, data balancing, and model selection significantly impact their performance.
- The proposed Random forest classification technique gave better accuracy and performance compared to the existing SVM

1.6 PROBLEM STATEMENT

The Existing machine learning algorithms are useful for detecting diseases but often lack accuracy due to their focus on preprocessing techniques, data balancing, and the use of various supervised and semi-supervised learning models. There is a need for a new technique that incorporates decision-level fusion to integrate the accuracy of multiple machine learning algorithms and achieve high disease detection accuracy. To address this, the paper proposes a fused machine learning model using Random forest algorithm and Artificial Neural Networks (ANN), combined with voting clasifier for decision-level fusion. This approach aims to improve the predictive accuracy of diabetes diagnosis by leveraging the strengths of Random Forest and ANN.

1.7 PROPOSED SYSTEM

The proposed system introduces a Fused Machine Learning Model (FMDP) for diabetes prediction, integrating ANN and RANDOM FOREST models. This fusion enhances prediction accuracy by leveraging the strengths of both algorithms. The system operates in two phases: training and testing. In the training phase, data is acquired, preprocessed, and classified using ANN and and Random Forest. The outputs of these models are then input into the Voting Classifier , which finalizes the prediction.In the proposed system we used feature scaling approach to improve the performance. This fusion approach has demonstrated a prediction accuracy of 98%, significantly improving over previous models.

The dataset used in the proposed system comes from the (University of California Irvine) machine Learning repository compiled by the hospital of Sylhet, Bangladesh. In the proposed system we are developing a web page by using HTML language and Flask to predict the diabetes postive or negative .

1.8 RESEARCH AIMS AND OBJECTIVES

This thesis aims to predict the diabetes positive or negative . To control the death ratio of diabetes.It can be controlled if the disease is diagnosed and preventative measures are taken in early-stage. By taking the symptoms of the patient in a real time and predict the diabetes by using fused machine learning approach.

1.9. Experimental Tool and Datasets

Jupyter Notebook

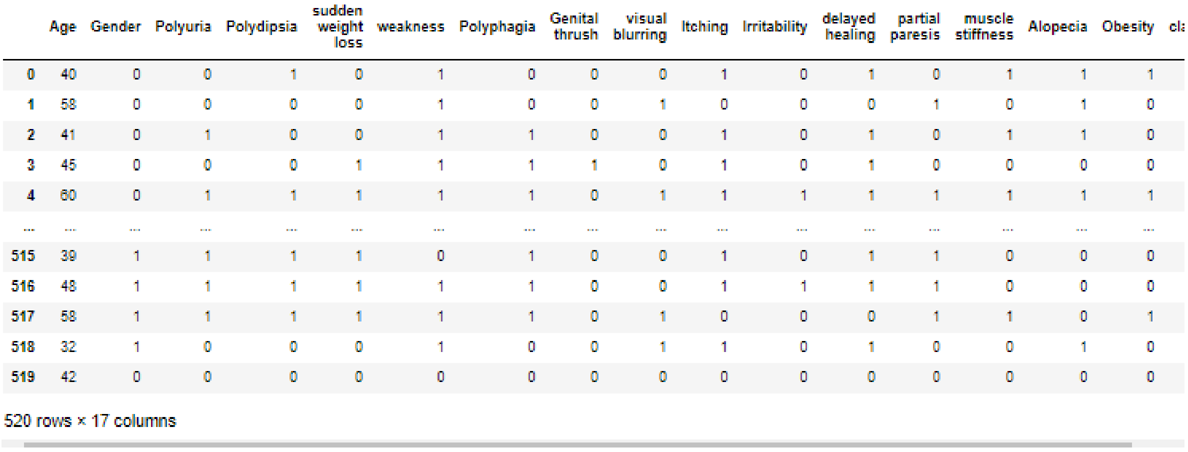
The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience. Jupyter Notebook supports many popular machine-learning libraries which can be easily loaded into your notebook. As a programmer, you can perform the following using Google Colab.

❖ Download and install Jupyter Notebook(anaconda)

❖ Create a new file ".ipynb"

❖ Write and execute code in Python

Dataset : The dataset used in the proposed system comes from the (University of California Irvine) machine Learning repository compiled by the hospitalofSylhet,Bangladesh.



	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopecia	Obesity	cl
0	40	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	
1	58	0	0	0	0	1	0	0	1	0	0	0	1	0	1	0	
2	41	0	1	0	0	1	1	0	0	1	0	1	0	1	1	0	
3	45	0	0	0	1	1	1	1	0	1	0	1	0	0	0	0	
4	60	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
515	39	1	1	1	1	0	1	0	0	1	0	1	1	0	0	0	
516	48	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	
517	58	1	1	1	1	1	1	0	1	0	0	0	1	1	0	1	
518	32	1	0	0	0	1	0	0	1	1	0	1	0	0	1	0	
519	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

520 rows × 17 columns

Fig 1.9 Dataset

1.10 STRUCTURE OF THESIS

The Thesis is organized accordingly:

Chapter 1: Introduction

This Chapter gives an overview of various concepts, such as diabetes prediction, machine larning technology, and ML in heathcare ,to get an insight into the topics that we deal with in the research work, required during our investigation and experimentation, and introduces relevant software that has been vastly used throughout the problem statement, Research objectives, the motivation for the research also discussed in this part.

Chapter 2: Literature Review

In this chapter, an in-depth discussion of various concepts that are needed for this research work is done. A detailed description of various existing systems is also discussed. A vast study of previous literature was conducted to identify the research gaps which is helpful in my research.

Chapter 3: System Analysis

In this chapter system requirements and specifications have been discussed.

Chapter 4: Modules Implementation

This chapter shows the detailed implementation of all the modules including Data collection,Data preprocessing and model evaluation and tuning and fusion.

Chapter 5: Source Code

This chapter consists of source code about the Training and Testing the data and model training and evalution and performance analysis part of the proposed system.

Chapter 6: Results

This chapter shows the experimental results of our proposed system.

Chapter 7 Conclusion and Future Scope

In this chapter, conclusions derived by the author in this research work are discussed along with the possible areas of research on related topics, and further studies are presented.

References: This chapter consists of all the references made to implement this

“Diabetes Prediction using Fused Machine Learning”.

CHAPTER-2

LITERATURE REVIEW

CHAPTER-2

LITERATURE REVIEW

2.1 LITERATURE REVIEW

[26] M. A. Sarwar, N. Kamal, W. Hamid, and M. A. Shah, This paper, used machine-learning algorithms for the detection of diabetes at an early stage by using Pima Diabetes dataset.In this two algorithms are used there are K nearest Neighbour and Support vector machine and other four algorithms to predict the diabetes .Their accuracy rates achieved from KNN and SVM were 77%, which is higher than the other four algorithms.A limitation of this paper is the size of the dataset and the missing values.

[27] Aiswarya Iyer, S. Jeyalatha and Ronak Sumbaly.This paper, used J48 Decision Tree and Naive Bayes approach for diagnosis of diabetes. Pima Indians Diabetes Database were used for diagnosis of diabetes with 768 Instances .Bayes gives least error rate and thus outperforms J48 decision Algorithm .Both the models are efficient in the diagnosis of diabetes using the percentage split of 70:30 of the data set. A developed model for diagnosis of diabetes will require more training data for creation and testing.Comparison of only 2 algorithm is not sufficient to build best diagnosis model.

[28] Quan Zou, Kaiyang Qu1 , Yamei Luo, Dehui Yin , Ying Ju and Hua Tang **.** Machine learning has been applied to many aspects of medical health. In this paper used decision tree, random forest and neural network to predict diabetes mellitus .we used Luzhou to represent the dataset from hospital physical examination data in Luzhou, China and Pima Indians represents the Pima Indians diabetics data. The two datasets contain 14 and 8 attributes, respectively. The random forest algorithm was obviously better than the other classifiers.The best result for Luzhou dataset is 80.84%, and the best performance for Pima Indians is 77.21%. Due to the taken data, we cannot predict the type of diabetes.

[29] Prof. Dhomse Kanchan B, Mr. Mahale Kishor M This paper used Diabetic patients dataset is collected from hospital repository with 1865 instances . Algorithms were implemented by using WEKA data mining technique to analyze algorithm accuracy. SVM and Naive Bayes applied with and without feature selection to predict diabetes disease using WEKA tool.Naive Bayes have better accuracy results and takes less time for building the training model than SVM with 34.89% . Classification Accuracy of Naive Bayes is 34.89% which is quite risky for prediction

[30] Sidong Wei , Xuejiao Zhao , Chunyan Miao This paper, we make a comprehensive exploration to the most popular techniques i.e., DNN (Deep Neural Network), SVM (Support Vector Machine) used to identify diabetes and data preprocessing methods.The experiment we carry out has four steps in total, the first step is to find the best data preprocessor for each classifier, while the next step is to optimize the parameters of each classifier. The third step is to compare these technique of diabetes identification by their accuracy.In this paper used the data set of Pima Indian women about their diabetes recognition. DNN algorithm performs the best and achieves the best accuracy of 77.86% .

[31] Samrat Kumar Dey, Ashraf Hossain , Md. Mahbubur Rahman . This paper used supervised learning algorithms for testing out accuracy among some sort of popular Machine Learning (ML) algorithms i.e., Support Vector Machine (SVM), K- Nearest Neighbors(KNN) , Naive Bayes Algorithm and Artificial Neural Network (ANN) .Artificial Neural Network (ANN) provide us highest accuracy of 82.35% with Min Max Scaling Method on Indian Pima Dataset.In this we have used PHP Web programming language as backend development, JavaScript frameworks as frontend and Tensorflow.js for the code implementation of Machine Learning Model.

[32] J.Lysa Eben , R.Jayasudha , S. Ramya , S.Kaliappan, Shobha Aswal This paper, used the Indian Pima Dataset for diabetes prediction .And Current techniques have a poor degree of precision in classification and forecast. The newer sample is superior to the older one based on categorization accuracy.After applying many machine-learning algorithms i.e., randomforest ,guassian , svc algorithms to the dataset. The best reliability, 97%, is achieved using logistic regression.

[33] Md. Faisal Faruque, Asaduzzaman ,Iqbal H. Sarker This paper,used the diagnostic dataset having 16 attributes diabetic of 200 patients. These attributes are age, diet, hyper-tension, problem in vision, genetic etc. Here used the most popular machine learning algorithms, namely Support Vector Machine (SVM), Naive Bayes (NB), K-Nearest Neighbor (KNN) and C4.5 decision tree (DT), on adult population data to predict diabetic mellitus. The classifier C4.5 achieves better results than other classifiers to predict diabetes mellitus. C4.5 achieves the better accuracy of 73.5% to predict diabetes mellitus utilizing a given medical dataset.

[34] Ayman Mir , Sudhir N. Dhage Healthcare domain is a very prominent research field n. Millions of patients seek treatments around the globe with various procedure . Machine Learning is a very promising approach which helps in early diagnosis of disease and might help the practitioners in decision making for diagnosis. This paper, used the Pima Indians Diabetes Database which is collected from National Institute of Diabetes .Here WEKA tool used to predict diabetes disease by employing Naive Bayes, Support Vector Machine, Random Forest and Simple CART algorithms. The overall performance of Support Vector machine to predict the diabetes disease is better than Naive Bayes, Random Forest and Simple Cart algorithms According to the Classification Accuracy of SVM is the highest which is 0.7913. The training time of Naive Bayes is less than SVM. The training time of Simple CART is the highest. Overall according to classification Accuracy SVM outperformed all other classifier.

[35] Kezban Alpan, Galip Savasilgi . this paper author [6] used data mining techniques to reveal relationships between data and make accurate predictions. Here the dataset which has been obtained from UCI machine learning depository contains 520 instances, each having 17 attributes. Weka tool used for diabetes prediction ,WEKA contains tools for tasks such as data pre-processing, clustering, classification, association rules, regression and visualization. In this paper, WEKA has been used as a data mining engineering This paper , we used Seven different classification algorithms i.e., Bayes Network, Naïve Bayes, J48, Random Tree, Random Forest, k-NN and SVM .And among the seven algorithms k-NN performed the highest accuracy with 98.07%.

[36] Shameem Hasan this paper,Author[10] used the datasets consist of several medical predictor variables and one target variable. Independent variables include the Body Mass Index (BMI), insulin level, age, number of pregnancies the patient had and some others. Based on these parameters, we can prediction the diabetes by applying artificial intelligence technique.Bayesian regularization, Levenberg–Marquardt algorithm and scaled conjugate are the various algorithmic parameters and functions used in the neural networks. The nftool (neural fitting tool) of MATLAB has been used in our proposed work to determine the performance and the results. In this paper, the predictive analysis of diabetes has been done through artificial neural networks. The different parameters of the neural network have been used to get the accurate result.

[37] Saloni Kumari , Deepika Kumar , Mamta Mittal this paper , author[11] used the Pima Indian Dataset for the experimentation The dataset has 9 columns and one output column with a dichotomous value to specify if the person has diabetes positive or diabetes negative. The proposed methodology uses ensemble of three machine learning models i.e., Random Forest, Logistic Regression, Naïve Bayes with soft voting classifier. Accuracy, precision, recall, F1-score, AUC curve have been taken as the evaluation criteria for testing the robustness of proposed methodology .The ensemble soft voting classifier has given 79.08% accurate results on the Pima Indians diabetes dataset.

[38] Soumayadeep Manna, Swagata Maity, Souvik Munshi, Mainak Adhikari This paper uses the classification and predictive analysis algorithm to predict the important factors for the cause of diabetes in the cloud environment. Here proposed a new analytical way of predicting Diabetes by using Cloud and Analytics.Pima Indian Diabetes dataset obtained from Kaggle Repository. The proposed model gives more accuracy with the non-medical factors. PIMA Indian Diabetes Dataset consists of all medical factors, but with the importance analysis, it shows only Glucose as an important medical term whereas all other non-medical factors have major priority.Logistic Regression , Random Forest algorithms are used for Diabetes Prediction Model Using Cloud Analytics. Logistic Regression is outperformed Random Forest with 86.7%. So used Logistic Regression as our model for paper.

[39] M. Tech. Scholar Arvind Aada , Prof. Sakshi Tiwari this paper, author[13] uses the Pima Indian diabetes database was obtained from UCI storehouse utilized for investigation. The R programming was utilized as digging device for diagnosing diabetes. R 3.5 is used for predicting the diabetes, PIMA Indian Diabetes Dataset from UCI storehouse contains 768 instances. The PIMA dataset is changed over from CSV to ".ARFF" group acknowledged by R 3.5. we have used different classifiers like Decision Trees, KNN and Naïve Bayes and ad boost .After Applying bootstrapping resembling method on this PIMA dataset will builds the exactness of practically all classifiers yet the choice trees leads over others with 94.44 % accuracy.

[40] Minyechil Alehegn , Rahul Joshi To group and predict symptoms in medical data, various data mining techniques were used by different researchers in different time. In this paper, Author[14] used PIDD (Pima Indian Diabetes Data Set) with 768 instances. Here we used KNN, Naïve Bayes, Random forest, and J48.In this study Weka 3.8.1 and java using netbean 8.2 used for analysis, classification, and prediction. And by using these algorithms [14] ensemble a hybrid model by combining individual techniques/methods into one in order to increase the performance and accuracy.A hybrid model provides best performance and accuracy than the single one

OBSERVATION FROM LITERATURE SURVEY:

In summary, these research papers used to predict the diabetes postive or negative to save several lives .In this they used so many algorithms and techniques to achieve high accuracy because the accuracy of the proposed models in disease prediction has always been the main concern of researchers.

As compared other techniques fused machine learning achives the higher accuracy Moreover, the death ratio of diabetes can be controlled if the disease is diagnosed and preventative measures are taken in early-stage.

CHAPTER 3

SYSTEM ANALYSIS

CHAPTER 3
SYSTEM ANALYSIS

3.1 PROPOSED SYSTEM

The proposed system introduces a Fused Machine Learning Model (FMDP) for diabetes prediction, integrating ANN and RANDOM FOREST models. This fusion enhances prediction accuracy by leveraging the strengths of both algorithms. The system operates in two phases: training and testing. In the training phase, data is acquired, preprocessed, and classified using ANN and and Random Forest. The outputs of these models are then input into the Voting Classifier , which finalizes the prediction.In the proposed system we used feature scaling approach to improve the performance. This fusion approach has demonstrated a prediction accuracy of 98%, significantly improving over previous models.

The dataset used in the proposed system comes from the (University of California Irvine) machine Learning repository compiled by the hospital of Sylhet, Bangladesh. In the proposed system we are developing a web page by using HTML language and Flask to predict the diabetes postive or negative .

ADVANTAGES

The proposed Fused Machine Learning Model (FMDP) for diabetes prediction offers several advantages, which are outlined below:

1. Enhanced Prediction Accuracy:

- By combining Artificial Neural Networks (ANN) and Random Forest models, the system leverages the strengths of both algorithms. ANN is proficient in capturing complex, non-linear relationships in the data, while Random Forest is robust against overfitting and can handle high-dimensional data effectively. The fusion of these models results in a significant improvement in prediction accuracy, achieving 98%.

2. Robustness and Reliability:

- The use of a Voting Classifier, which combines the outputs of ANN and Random Forest, enhances the robustness of the system. This approach reduces the risk of individual model biases and ensures that the final prediction is more reliable and stable.

3. Improved Performance with Feature Scaling:

- Implementing feature scaling ensures that the input features contribute equally to the model training process. This technique helps in speeding up the convergence of the learning algorithms and enhances the overall performance of the models.

4. Comprehensive Training and Testing Phases:

- The system follows a structured approach with distinct training and testing phases. In the training phase, data is acquired, preprocessed, and classified, ensuring that the models are well-trained on diverse data. This rigorous training phase improves the generalization capability of the models when applied to new, unseen data during the testing phase.

5. Utilization of a Reliable Dataset:

- The dataset from the University of California Irvine (UCI) Machine Learning Repository, compiled by the hospital of Sylhet, Bangladesh, is known for its quality and reliability. Using a well-established dataset enhances the credibility of the model's predictions.

6. Web-Based Interface for Accessibility:

- Developing a web page using HTML and Flask to predict diabetes status makes the system user-friendly and easily accessible. This web-based interface allows users, including healthcare professionals and patients, to input data and receive predictions in a convenient manner.

7. Potential for Real-Time Predictions:

- With the web-based system, the proposed model has the potential to provide real-time predictions, which can be crucial for timely medical interventions and decision-making.

8. Scalability and Flexibility:

- The system's architecture, incorporating widely used machine learning models and web technologies, ensures that it can be easily scaled and adapted for other medical prediction tasks or expanded to include additional features and functionalities.

Overall, the proposed FMDP system represents a significant advancement in diabetes prediction by integrating advanced machine learning techniques, ensuring high accuracy, and providing a practical, user-friendly tool for healthcare applications.

3.2PROPOSED SYSTEM ARCHITECTURE

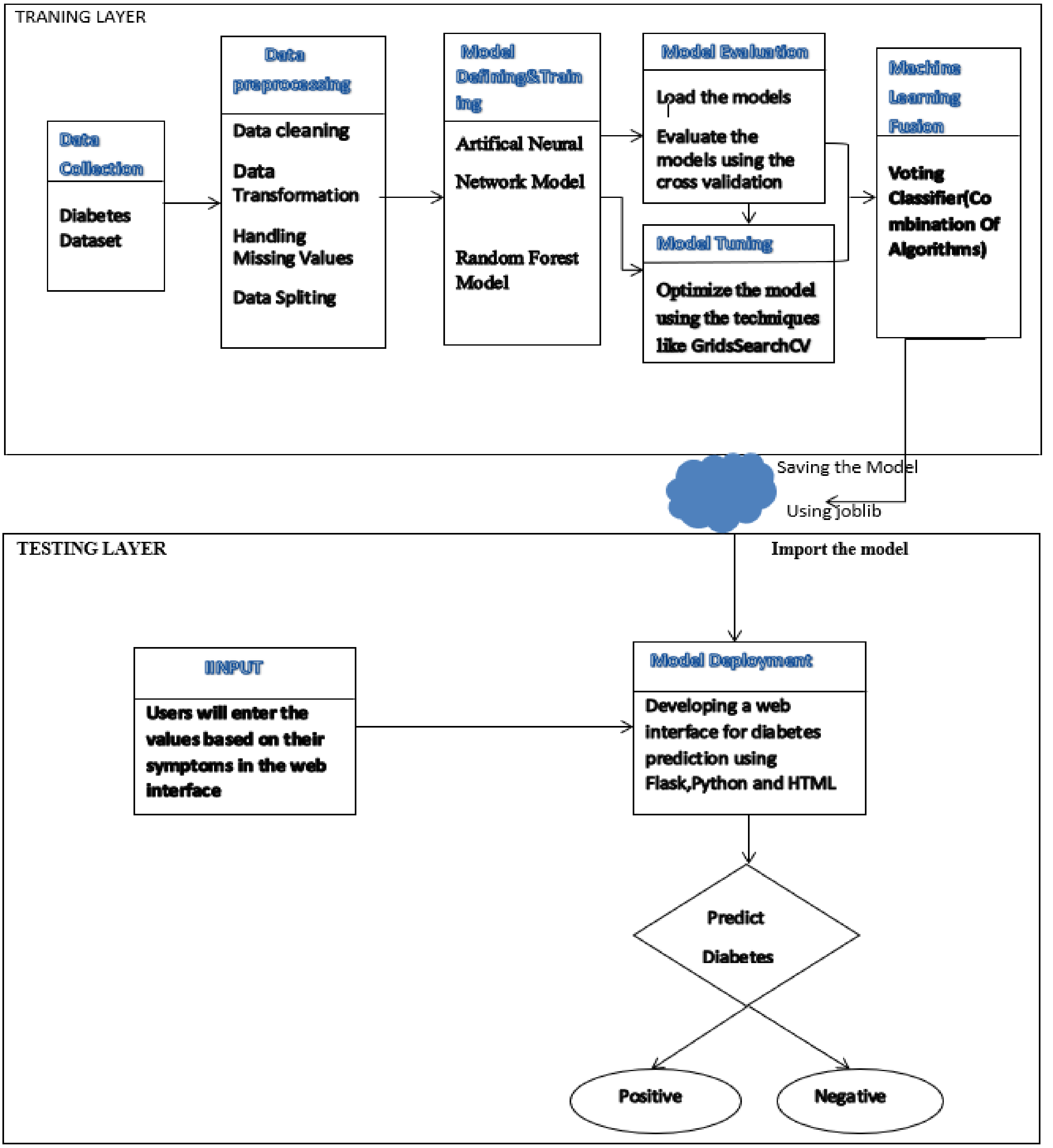


Fig 3.1 Architecture of the proposed system

3.3 MODULES

3.3.1 Data Collection :

Dataset: The diabetes dataset is obtained from the University of California Irvine (UCI) Machine Learning Repository, compiled by the hospital of Sylhet, Bangladesh.

3.3.2 Data Preprocessing :

- Data Cleaning: Removing or correcting any errors or inconsistencies in the dataset.
- Data Transformation: Converts categorical variables to numerical values for easier processing and analysis
- Handling Missing Values: Managing any missing data points by either filling them in or removing them.
- Data Splitting: Dividing the dataset into training and testing sets to evaluate.

3.3.3 Model Defining & Training :

Artificial Neural Network (ANN) Model: Define and train the ANN model to capture complex relationships in the data.

Random Forest Model: Define and train the Random Forest model to leverage its robustness and handle high-dimensional data.

3.3.3.1 ARTIFICIAL NEURAL NETWORK ALGORITHM

Artificial Neural Network (ANN):

Type: Deep learning model inspired by the human brain.

Objective: Used for various tasks including classification, regression, and more complex tasks like image and speech recognition.

Operation:

Composed of interconnected nodes (neurons) organized in layers (input, hidden, output).

- Each connection between nodes has a weight that is adjusted during training to minimize the error between predicted and actual output.

Advantages:

- Can capture complex relationships in data.
- Effective for tasks with large amounts of data.

Uses and applications

- Artificial Neural Networks (ANNs) are widely used in various fields due to their ability to model complex patterns and relationships in data. Here are some of the primary uses and applications of ANNs:

➢ **Healthcare**

❖ **1. Disease Diagnosis:**

- ANNs are used to predict diseases such as diabetes, cancer, and heart disease by analyzing medical data and patient records.

❖ **2. Medical Imaging:**

- They help in image classification tasks like identifying tumors in radiology scans, segmenting medical images, and enhancing image quality.

❖ **3. Drug Discovery:**

- ANNs assist in predicting the effects of new drugs, optimizing drug formulations, and analyzing chemical compounds.
- ANNs detect fraudulent transactions by identifying unusual patterns and behaviors in financial data.

The versatility and ability of ANNs to learn from data make them invaluable across a wide range of applications, driving innovation and efficiency in various industries.

3.3.3.2 RANDOM FOREST :

Random Forest:

Type: Ensemble learning method.

Objective: Used for classification and regression tasks.

Operation:

- Constructs multiple decision trees during training.
- Each tree is trained on a random subset of the data (bootstrap sample) and a random subset of features.
- During prediction, aggregates the predictions of all trees to make a final prediction (mode for classification, average for regression).

Advantages:

- Handles high-dimensional data well.
- Less prone to overfitting compared to individual decision trees.

Uses and Advantages:

Random Forest is a versatile and widely used machine learning algorithm due to its robustness, accuracy, and ease of use. It is an ensemble learning method that combines multiple decision trees to improve predictive performance and control overfitting. Here are some of the primary uses and applications of Random Forest:

Healthcare

1. **Disease Diagnosis**:
    - Random Forest models are used to predict the presence of diseases such as diabetes, cancer, and heart disease by analyzing patient data.
2. **Genomic Studies**:
    - They help in identifying gene expressions and genetic markers associated with specific diseases.
3. **Medical Imaging**:
    - Random Forest algorithms are used to classify and segment medical images for identifying abnormalities.

The flexibility and robustness of Random Forest make it suitable for a wide range of applications, driving insights and decision-making across various domains.

3.3.4 Machine Learning Fusion:

Voting Classifier: Combine the outputs of ANN and Random Forest models using a Voting Classifier to finalize the prediction.

A Voting Classifier is an ensemble machine learning model that combines the predictions from multiple different models (or classifiers) to make a final prediction. The idea is to leverage the strengths of each individual model to improve overall performance and accuracy. Voting classifiers can be particularly useful in situations where different models perform well on different aspects of the data.

Types of Voting

There are two main types of voting mechanisms in a Voting Classifier:

1. **Hard Voting**:
    - In hard voting, the final prediction is made based on the majority vote from all the individual models.
    - Each model votes for a class, and the class that receives the most votes is chosen as the final prediction.
    - For example, if three models predict classes as follows: Model 1 predicts A, Model 2 predicts B, and Model 3 predicts B, the final prediction will be B, as it has the majority vote.
2. **Soft Voting**:
    - In soft voting, the final prediction is made based on the average of the predicted probabilities from all the individual models.
    - Each model outputs a probability for each class, and these probabilities are averaged. The class with the highest average probability is chosen as the final prediction.

- For example, if three models predict probabilities for class A as follows: Model 1 predicts 0.2, Model 2 predicts 0.4, and Model 3 predicts 0.6, the average probability for class A is (0.2 + 0.4 + 0.6) / 3 = 0.4. The class with the highest average probability across all models is chosen.

3.3.4.1 WORKING OF VOTING CLASSIFIER

Model Selection:

- Choose a diverse set of base models (e.g., logistic regression, decision trees, support vector machines, neural networks). The diversity in models helps to capture different patterns and reduce the risk of correlated errors.

Training:

- Train each of the selected base models on the training data independently.

Prediction:

- For a new input instance, each base model makes a prediction (either a class label or probability).
- In hard voting, the class labels are combined, and the majority class is selected.
- In soft voting, the predicted probabilities are averaged, and the class with the highest average probability is selected.

3.3.4.2 Advantages of Voting Classifier

1. **Improved Performance**:
   - By combining the predictions of multiple models, a Voting Classifier can achieve better performance than any individual model, especially when the individual models are diverse.
2. **Reduced Overfitting**:

   - Ensemble methods like voting help to reduce overfitting by averaging out the errors of individual models, making the final model more robust.

3. **Flexibility**:
   - Voting Classifiers can be constructed with any combination of base models, making them highly flexible and adaptable to various types of data and problems.

3.3.4.3 Applications of Voting Classifier

1. **Healthcare**:
   - Predicting disease outcomes by combining models trained on different features such as clinical data, imaging data, and genetic data.
2. **Finance**:
   - Credit scoring and fraud detection by combining models that analyze different aspects of financial data.
3. **Marketing**:
   - Customer segmentation and churn prediction by combining models that consider various customer behaviors and attributes.
4. **Image and Text Classification**:
   - Improving accuracy in image and text classification tasks by combining models with different architectures and feature extraction methods.

Overall, Voting Classifiers are powerful tools that enhance predictive performance and reliability by leveraging the strengths of multiple models.

3.3.5 Model Evaluation :

- **Load the Models**: Load the trained models for evaluation.

- **Cross Validation**: Evaluate the models using cross-validation techniques to ensure reliability and generalizability.

3.3. 6 Model Tuning:

- **Optimization**: Optimize the model parameters using techniques like GridSearchCV to enhance performance.
- **Saving the Model**: Save the optimized model for future use.

3.3.7 Testing Layer

1. **Import the Model**:
    - **Using joblib**: Import the saved model for deployment.
2. **Input**:
    - **User Interface**: Users will enter their symptoms through a web interface developed using Flask and HTML.
3. **Model Deployment**:
    - **Web Interface**: Develop a user-friendly web page for diabetes prediction.
4. **Prediction**:

    1. **Diabetes Status**: The model will predict whether the user is positive or negative for diabetes based on the input values.

These modules collectively form a comprehensive system for accurate and user-friendly diabetes prediction.

3.4 SOFTWARE REQUIREMENTS

- Operating Systems: windows 8/10/11,Ubuntu,mac
- Programming Language: Python, HTML
- Version: 3.9.0
- Softwares Required: MATLAB R2020a/jyupter

3.4.1 OPERATING SYSTEM

In Python, the string type is used to represent various elements like file names, command-line arguments, and environment variables. In some situations, these strings need to be encoded and decoded to transition between text and bytes before interacting with the underlying operating system.

Python employs the encoding specified by the file system to perform this transformation. This flexibility allows developers to work with OS-specific functionality effectively. The "fileinput" module helps read lines from files specified on the command line, while the "os.path" module aids in path manipulation. Furthermore, the "open()" function simplifies reading from or writing to files.

For creating temporary files and directories, it's recommended to use the "tempfile" module, while the "shutil" module is more suitable for performing advanced file and directory operations. Python's built-in modules that rely on OS-specific features maintain a consistent interface when a particular capability is universally accessible. For example, the "os.stat(path)" function provides uniform stat data for the specified path, aligning with the POSIX interface. However, be cautious when using OS-specific extensions from the "os" module, as they may compromise cross-platform compatibility.

Every function that deals with paths or file names can handle both bytes and string objects, producing output of the same data type. It's worth noting that certain functionalities like "os.fork," "os. exec," and "os.spawn*p*" are not supported on the VxWorks platform. In the development of this project, the Windows operating system was used.

3.4.2 INTRODUCTION TO PYTHON

Python, an open-source high-level programming language, was conceptualized by Guido van Rossum in the late 1980s and is currently maintained by the Python Software Foundation. It evolved from the ABC language, a precursor created during the early stages of Rossum's career. Python stands as a robust language enabling the development of games, graphical user interfaces (GUIs), and web applications. Its characteristic feature lies in its readability, allowing developers to craft code that resembles natural English statements. Python programs need processing before machines can execute them due to their human-readable nature. Being an interpreted language, Python translates code into machine-readable byte code each time a program runs. Furthermore, Python embraces an object-oriented approach, affording users control over data structures or objects for program creation and execution. In Python, all elements, including objects, data types, functions, methods, and classes, hold equal importance, representing a first-class paradigm in the language.

Python has stood the test of time and remains pertinent in various industries, businesses, and among individual programmers and users. Its endurance and versatility stem from its ability to enhance productivity, communication, and efficiency in various domains. As a highly recommended first programming language, Python continues to thrive and contribute to the programming landscape, embodying a living, thriving, and exceedingly valuable language.

HTML

HTML (HyperText Markup Language) is the standard language used to create and structure content on the web. It uses a system of tags to define elements within a document, enabling the creation of web pages that can be displayed in web browsers. Here are the key points about HTML:

Basic Structure

An HTML document has a basic structure consisting of elements defined by tags. The fundamental tags include:

- <!DOCTYPE html>: Declares the document type and HTML version.
- <html>: The root element of an HTML page.
- <head>: Contains metadata, links to stylesheets, and the title of the document.
- <title>: Sets the title displayed in the browser's title bar or tab.
- <body>: Contains the content of the document, such as text, images, links, and other elements.

Common HTML Elements

1. **Headings**: Defined by <h1> to <h6>, with <h1> being the highest level.
2. **Paragraphs**: Defined by <p>, used for blocks of text.
3. **Links**: Defined by <a href="URL">, used to create hyperlinks.
4. **Images**: Defined by <img src="URL" alt="description">, used to embed images.
5. **Lists**: Ordered lists (<ol>) and unordered lists (<ul>) contain list items (<li>).
6. **Tables**: Defined by `<table>`, `<tr>` (table row), `<td>` (table cell), and `<th>` (table header).

Forms

Forms are used to collect user input and submit it to a server:

- <form>: Container for form elements.
- <input>: Defines input fields (e.g., text, password, submit).
- <textarea>: Multi-line text input.
- <select>: Drop-down list.
- <button>: Clickable button.

Semantic HTML

Semantic HTML elements provide clear meaning to the content they contain:

- <header>: Defines a header section.
- <nav>: Defines navigation links.
- <section>: Defines a section of content.
- <article>: Defines an independent, self-contained article.
- <aside>: Defines content aside from the main content.
- <footer>: Defines a footer section.

HTML is essential for web development, providing the structure for web pages. It is complemented by CSS for styling and JavaScript for interactivity, forming the foundation of modern web development.

Jupyter Notebook

The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience. Jupyter Notebook supports many popular machine-learning libraries which can be easily loaded into your notebook. As a programmer, you can perform the following using Google Colab.

❖ Download and install Jupyter Notebook(anaconda)

❖ Create a new file ".ipynb"

❖ Write and execute code in Python

3.5 LIBRARIES

3.5.1 Anonypy

Scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3- Clause BSD license.

The project was started in 2007 by David Cournapeau as a Google Summer of Code project, and since then many volunteers have contributed. See the About us page for a list of core contributors.

The main use of Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction.

Installation

The easiest way to install scikit-learn is using pip:

```
pip install -U scikit-learn
```

3.5.2 Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram, etc.

Installation:

Windows, Linux, and macOS distributions have matplotlib and most of its dependencies as wheel packages. Run the following command to install the matplotlib package :

- **python -mpip install -U matplot**

3.6 HARDWARE REQUIREMENTS

- CPU: Intel core i3
- GPU: 2GB Graphic Card
- Storage: 256(SSD) & 1TB(HDD)
- RAM: 8GB

CHAPTER - 4

SYSTEM DESIGN

CHAPTER – 4

SYSTEM DESIGN

4.1 UML DIAGRAMS

UML, which represents Unified Modeling Language, is a standardized modeling language utilized in the domain of object-oriented software engineering.

The aim is to establish UML as a universal means for generating models of object-oriented computer software. The existing structure of UML consists of two main constituents: a Meta-model and a notation system. It's plausible that in the future, a method or procedure could be integrated into or linked with UML.

UML, which stands for Unified Modeling Language, is a universally accepted language used to define, visualize, build, and document software system components.

GOALS

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.

- ### 4.1.1 SEQUENCE DIAGRAM

Illustrates the interactions between objects or components in a system over time, showing the order in which messages are exchanged.

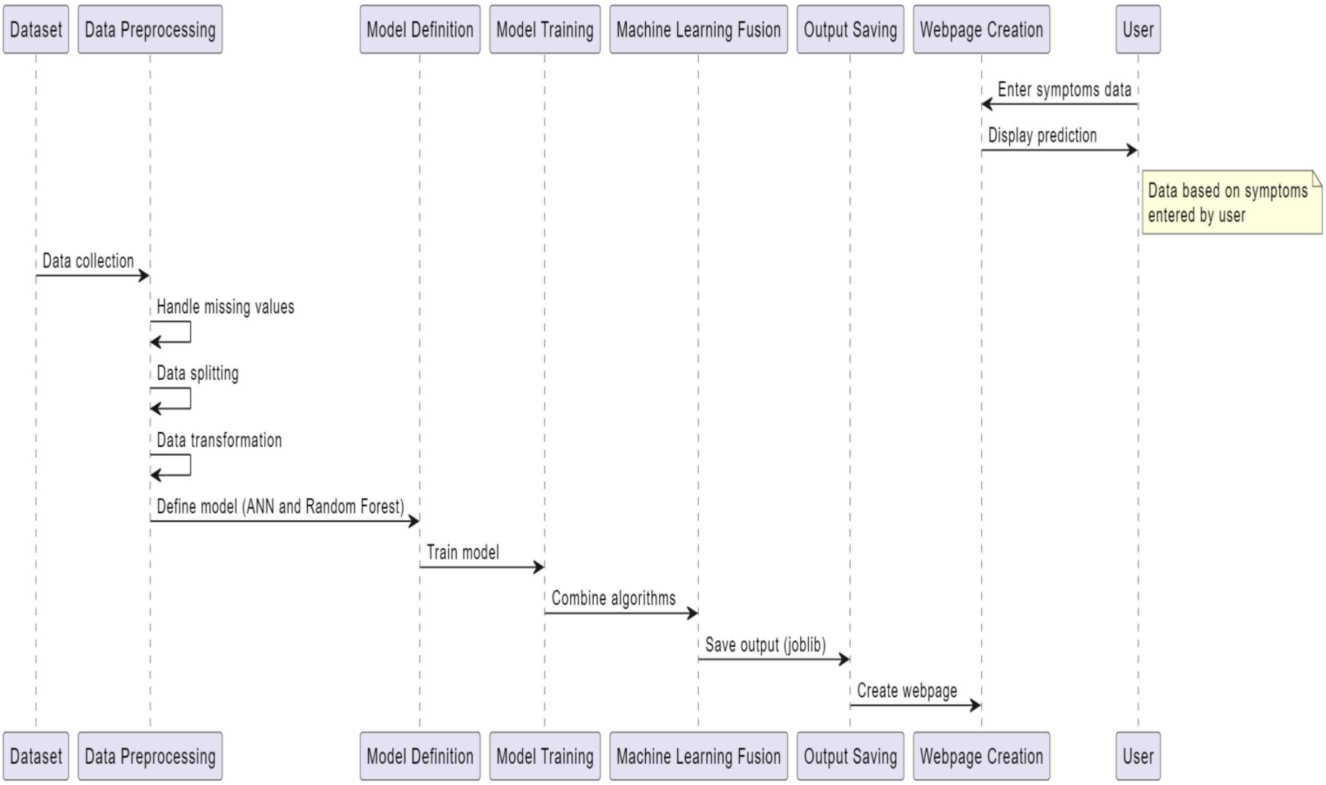


Fig 4.1 Sequence Diagram for proposed system

4.1.2 Class Diagram

Describes the structure of a system by showing the classes, their attributes, methods, and the relationships between them.

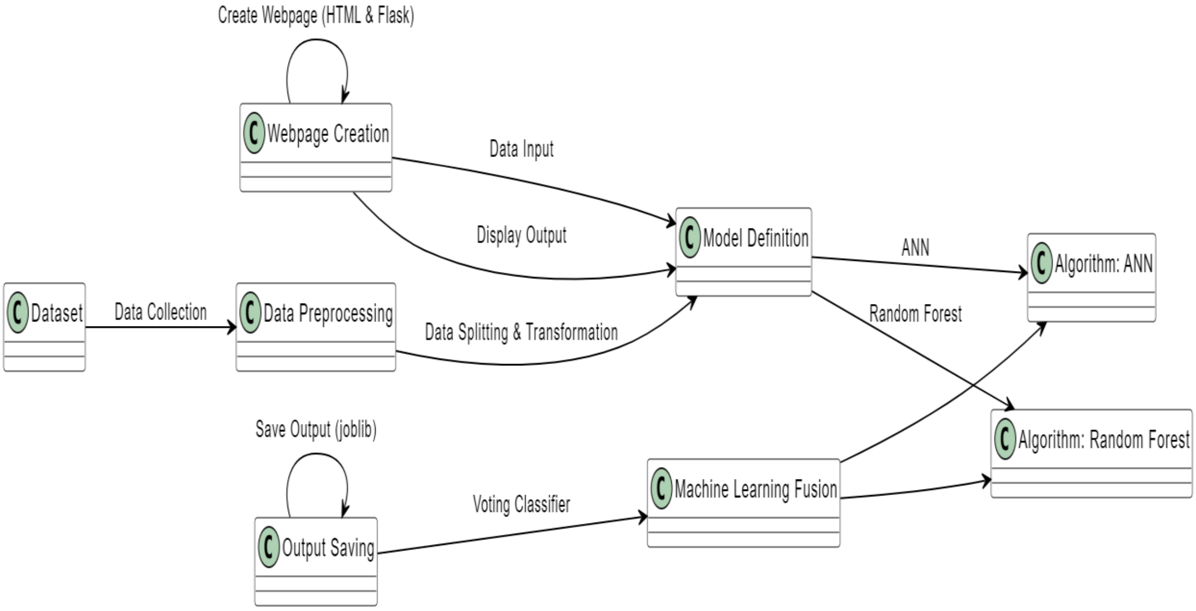


Fig 4.2 Class Diagram for proposed system

4.1.3 Use Case Diagram:

Represents the functionality of a system from the perspective of external entities (actors) and how the system responds to those interactions

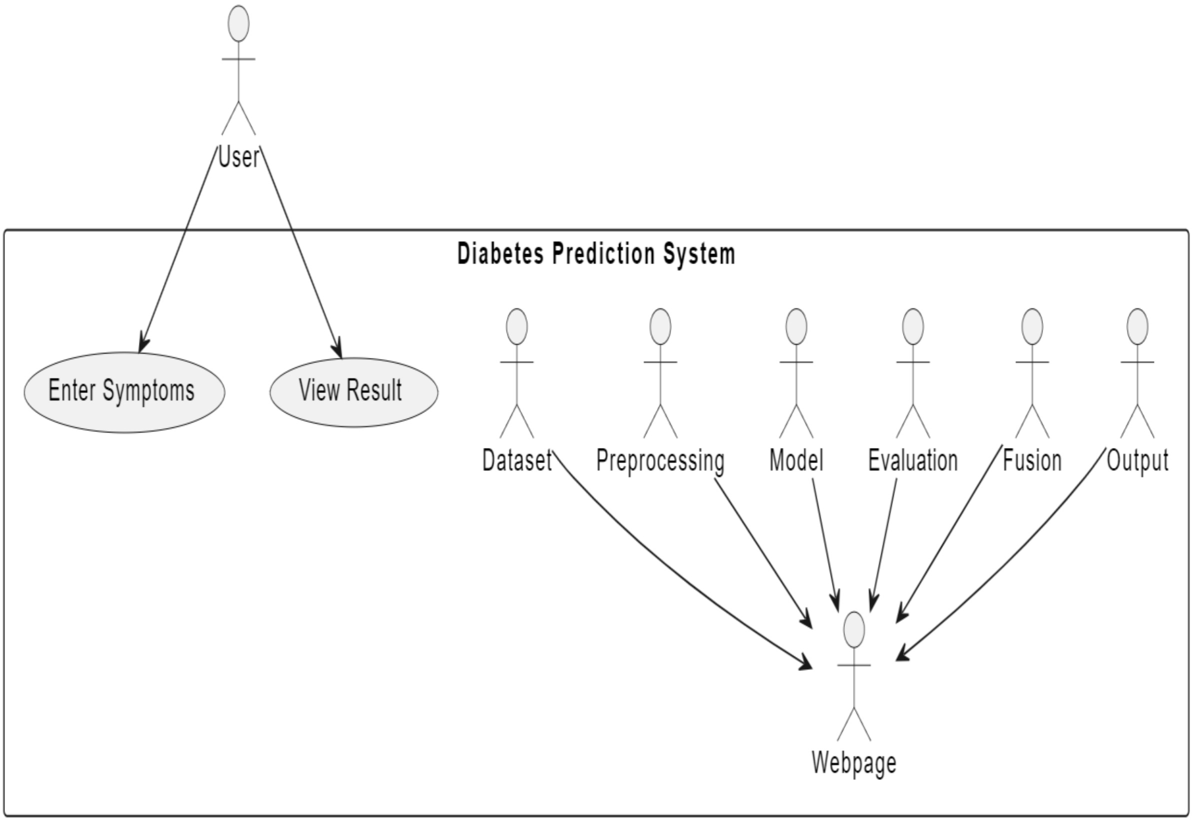


Fig 4.3 Use Case Diagram for proposed system

CHAPTER-5

IMPLEMENTATION

CHAPTER-5

IMPLEMENTATION

5.1 SETTING UP THE SYSTEM

We are implementing our project through Jyputer Notebook. So first we need to download and install the required software for the code to be implemented. We need to install software like Jyputer Notebook or MATLAB20a for the implementation of the project.

5.2 IMPORTING LIBRARIES

Let us also import the basic libraries that required for the project. Further, I will cover future imports depending on the model.

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score

from sklearn.preprocessing import StandardScaler

from sklearn.neural_network import MLPClassifier

from sklearn.ensemble import VotingClassifier, RandomForestClassifier

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, classification_report

import pickle
```

5.3 DATA COLLECTION

Download the datasets related to DIABETES prediction . And read the dataset

```
# Load dataset using Pandasy
```

```
data = pd.read_csv('diabetes.csv')

# display sample dataset

data.head()
```

5.4 DATA PREPROCESSING

5.4.1 Handling missing values

```
data.isnull().sum()

data.fillna(data.mean(), inplace=True)
```

5.4.2 Data Transformation

```
data = data.replace('No',0)

data = data.replace('Yes',1)

data = data.replace('Female',1)

data = data.replace('Male',0)

data = data.replace('Negative',0)

data = data.replace('Positive',1)

data
```

5.4.3 Data Spilting

```
from sklearn.model_selection import train_test_split

xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.3,random_state=42)
```

5. 5 MODEL DEFINING AND TRAINING

Artifical Neural Network

```
# ANN

ann = MLPClassifier(max_iter=1000)

ann_params = {'hidden_layer_sizes': [(10,), (50,), (100,)], 'activation': ['tanh', 'relu'], 'solver': ['sgd', 'adam']}

ann_grid = GridSearchCV(ann, ann_params, cv=5)
```

```
ann_grid.fit(xtrain, ytrain)
```

Random Forest

```
rf = RandomForestClassifier(random_state=42)
rf_params = {
    'n_estimators': [50, 100],  # Number of trees in the forest
    'max_depth': [10, 20, 30],  # Maximum depth of the tree
    'min_samples_split': [2, 5],  # Minimum number of samples required to split an internal node
    'min_samples_leaf': [1, 2],  # Minimum number of samples required to be at a leaf node
    'bootstrap': [True]  # Method for bootstrapping samples
}
rf_grid = GridSearchCV(estimator=rf, param_grid=rf_params, cv=5, n_jobs=-1)  # n_jobs=-1 for using all available processors
rf_grid.fit(xtrain, ytrain)  # Fit the model
```

5. 6 Model Evaluation

```
print("ANN Best Parameters:", ann_grid.best_params_)
print("Random Forest Best Parameters:", ann_grid.best_params_)
ann_cv_score = cross_val_score(ann, xtrain, ytrain, cv=5)
rf_cv_score = cross_val_score(rf, xtrain, ytrain, cv=5)
print("Best ANN Score:", ann_grid.best_score_)
print("Best Random Forest Score:", rf_grid.best_score_)
```

Evaluating with cross-validation

```
ann_cv_score = cross_val_score(ann_grid.best_estimator_, xtrain, ytrain, cv=5)

print(f"Cross-Validation Scores for ANN: {ann_cv_score}")

rf_cv_score = cross_val_score(rf_grid.best_estimator_, xtrain, ytrain, cv=5)

print(f"Cross-Validation Scores for Random Forest: {rf_cv_score}")
```

5.7 Machine Learning Fusion

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.ensemble import VotingClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.neural_network import MLPClassifier

from sklearn.pipeline import make_pipeline

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, precision_score, recall_score, f1_score

import joblib

# Assuming xtrain is a NumPy array, convert it to DataFrame

# Replace 'feature1', 'feature2', ..., 'feature16' with your actual column names

column_names = ['Age', 'Gender', 'Polyuria', 'Polydipsia','sudden weight loss','weakness','Polyphagia','Genital thrush', 'visual blurring','Itching','Irritability','delayed healing','partial paresis','muscle stiffness', 'Alopecia','Obesity']

xtrain_df = pd.DataFrame(xtrain, columns=column_names)

xtest_df = pd.DataFrame(xtest, columns=column_names)


# Define preprocessing steps for numerical and categorical features
```

```
numeric_features = xtrain_df.select_dtypes(include=['int64', 'float64']).columns
categorical_features = xtrain_df.select_dtypes(include=['object']).columns
numeric_transformer = StandardScaler()
categorical_transformer = OneHotEncoder(handle_unknown='ignore')
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)
    ]
)
# Create individual models
ann = MLPClassifier(max_iter=1000)
rf = RandomForestClassifier(random_state=42)
# Create a voting classifier
voting_clf = VotingClassifier(estimators=[
    ('ann', ann),
    ('rf', rf),
], voting='hard')  # You can choose 'hard' or 'soft' voting strategy
# Create a pipeline with preprocessing and model
pipeline = make_pipeline(preprocessor, voting_clf)
# Fit the pipeline
pipeline.fit(xtrain_df, ytrain)
# Save the fitted pipeline to a file
joblib.dump(pipeline, 'fused_model.pkl')
# Evaluate the pipeline
y_pred = pipeline.predict(xtest_df)
```

```
# Print metrics separately
accuracy = accuracy_score(ytest, y_pred)
precision = precision_score(ytest, y_pred, average='weighted')
recall = recall_score(ytest, y_pred, average='weighted')
f1 = f1_score(ytest, y_pred, average='weighted')
print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
print(f'F1 Score: {f1}')
print(f'Confusion Matrix:\n {confusion_matrix(ytest, y_pred)}')
print(f'Classification Report:\n {classification_report(ytest, y_pred)}')
```

5.8 Model Deployement

Flask code for web page implementation

```
from flask import Flask, request, render_template, jsonify
import joblib
import numpy as np
app = Flask(__name__)
# Load the pre-trained machine learning model
try:
    fused_model = joblib.load('fused_model.pkl')
except Exception as e:
    print(f"Error loading model: {str(e)}")
@app.route('/')
def index():
    return render_template('index.html')
```

```
@app.route('/submit', methods=['POST'])
def submit():
    try:
        # Parse form data
        Age = int(request.form['Age'])
        Gender = int(request.form['Gender'])
        Polyuria = int(request.form['Polyuria'])
        Polydipsia = int(request.form['Polydipsia'])
        sudden_weight_loss = int(request.form['sudden_weight_loss'])
        weakness = int(request.form['weakness'])
        Polyphagia = int(request.form['Polyphagia'])
        Genital_thrush = int(request.form['Genital_thrush'])
        visual_blurring = int(request.form['visual_blurring'])
        Itching = int(request.form['Itching'])
        Irritability = int(request.form['Irritability'])
        delayed_healing = int(request.form['delayed_healing'])
        partial_paresis = int(request.form['partial_paresis'])
        muscle_stiffness = int(request.form['muscle_stiffness'])
        Alopecia = int(request.form['Alopecia'])
        Obesity = int(request.form['Obesity'])
 # Prepare input data as numpy array
        input_data = [Age, Gender, Polyuria, Polydipsia, sudden_weight_loss,
weakness,Polyphagia, Genital_thrush, visual_blurring, Itching, Irritability,
delayed_healing, partial_paresis, muscle_stiffness, Alopecia, Obesity]
        input_data = np.array(input_data).reshape(1, -1)
```

```
# Perform prediction
  prediction = fused_model.predict(input_data)[0]
# Convert prediction to human-readable format
    result_label = "positive" if prediction == 1 else "negative"
 # Render a new HTML page with the prediction result
    return render_template('result.html', prediction=result_label)
     except KeyError as e:
  return jsonify({'error': f'Missing form field: {str(e)}'}), 400
    except ValueError as e:
        return jsonify({'error': f'Invalid input: {str(e)}'}), 400
    except Exception as e:
        return jsonify({'error': f'Prediction error: {str(e)}'}), 500
if __name__ == '__main__':
    app.run(debug=True)
```

CHAPTER – 6
RESULTS

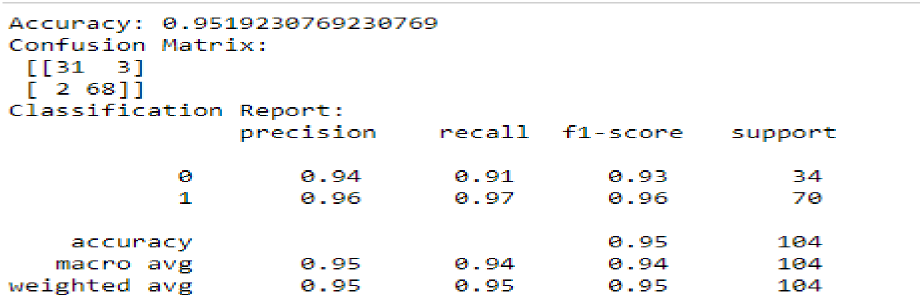
CHAPTER-6

RESULTS

The evaluation of the overall process is based on classification accuracy, precision, recall, and F1score. For comparison analysis, the research is conducted on three more standard classifiers namely ANN and Random forest.

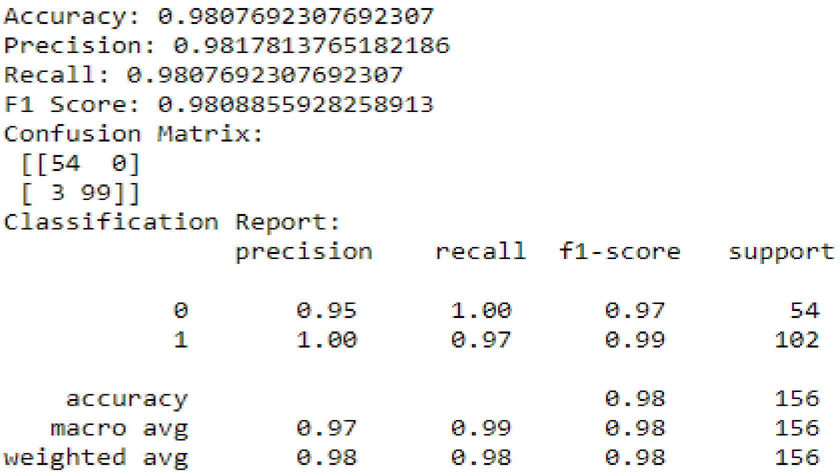
The proposed system has used (University of California Irvine) machine Learning repository compiled by the hospital of Sylhet, Bangladesh.

6.1. Classification Accuracy (%)



```
Accuracy: 0.9519230769230769
Confusion Matrix:
 [[31  3]
 [ 2 68]]
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.91      0.93        34
           1       0.96      0.97      0.96        70

    accuracy                           0.95       104
   macro avg       0.95      0.94      0.94       104
weighted avg       0.95      0.95      0.95       104
```

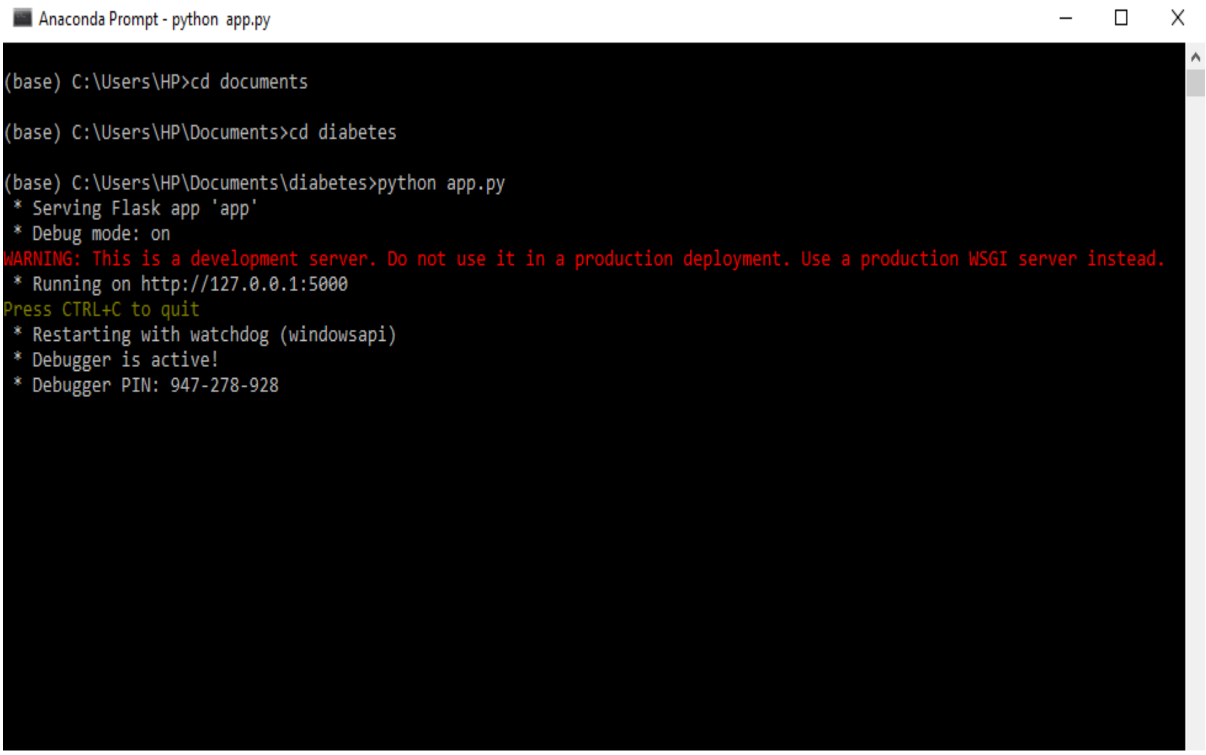
Fig 6.1.1 classification accuracy using SVM and ANN



```
Accuracy: 0.9807692307692307
Precision: 0.9817813765182186
Recall: 0.9807692307692307
F1 Score: 0.9808855928258913
Confusion Matrix:
 [[54  0]
 [ 3 99]]
Classification Report:
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        54
           1       1.00      0.97      0.99       102

    accuracy                           0.98       156
   macro avg       0.97      0.99      0.98       156
weighted avg       0.98      0.98      0.98       156
```

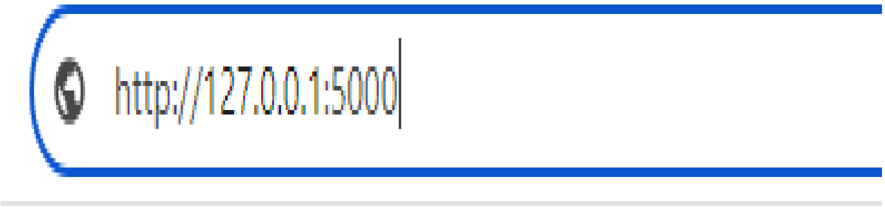
Fig 6.1.2 clssification accuracy using RF and ANN

The following method which proves the proposed method achieved better accuracy.



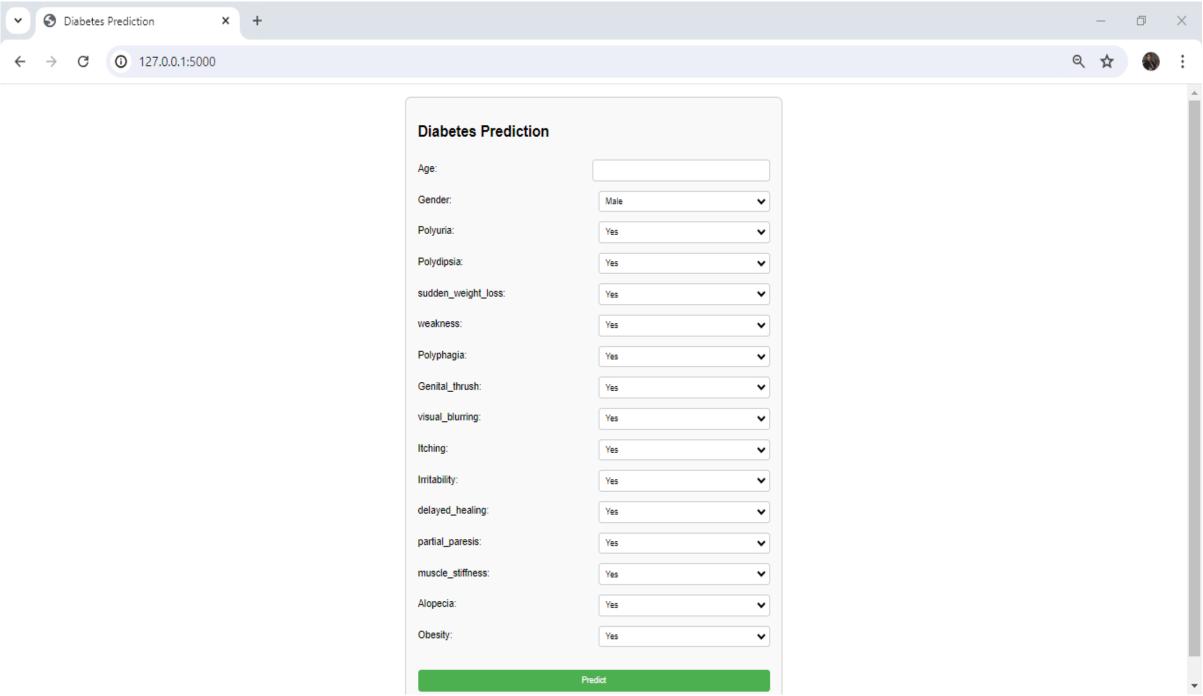
```
(base) C:\Users\HP>cd documents

(base) C:\Users\HP\Documents>cd diabetes

(base) C:\Users\HP\Documents\diabetes>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 947-278-928
```

Fig 6.2 In above screen python server started and now open browser and enter URL as ‘http://127.0.0.1: 5000/index.html’



http://127.0.0.1:5000

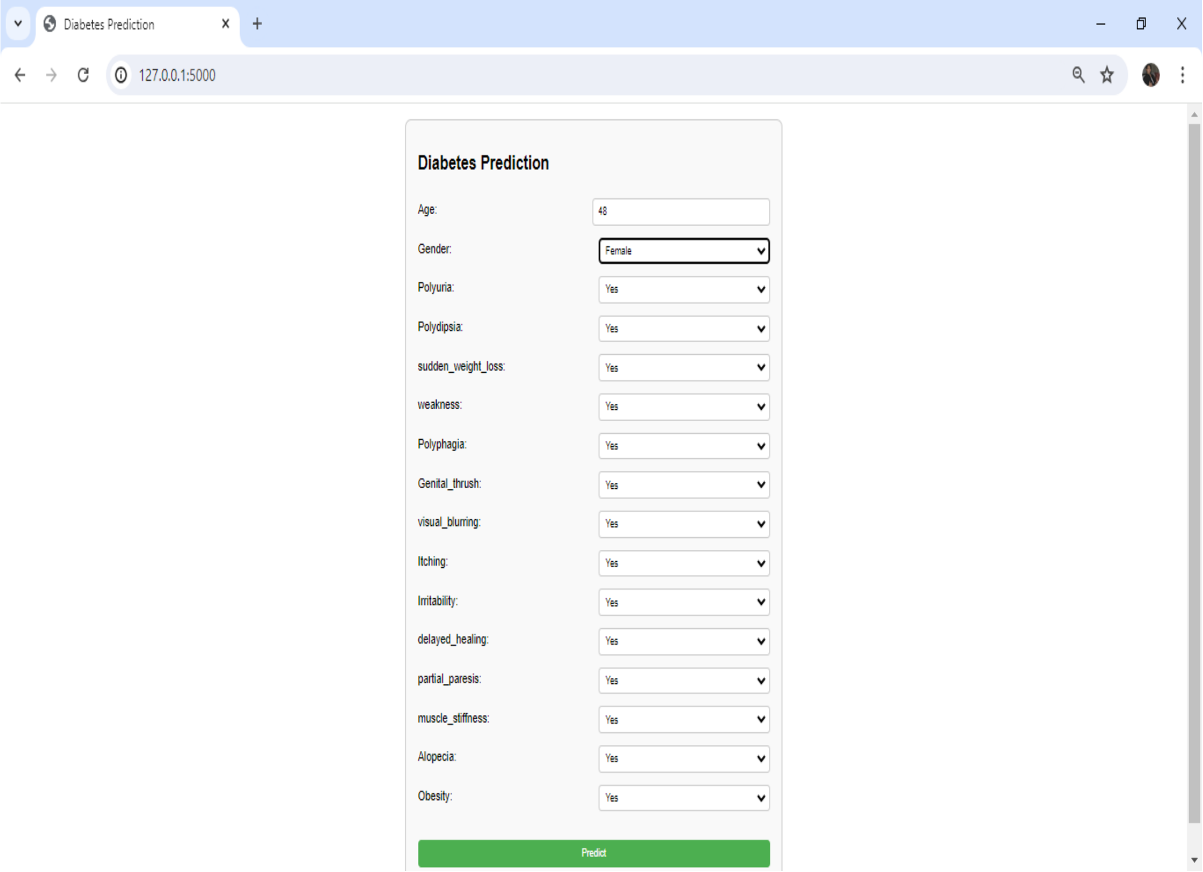
Fig 6.3 paste the link on the web browser and click on it



Diabetes Prediction

Age:
Gender: Male
Polyuria: Yes
Polydipsia: Yes
sudden_weight_loss: Yes
weakness: Yes
Polyphagia: Yes
Genital_thrush: Yes
visual_blurring: Yes
Itching: Yes
Irritability: Yes
delayed_healing: Yes
partial_paresis: Yes
muscle_stiffness: Yes
Alopecia: Yes
Obesity: Yes

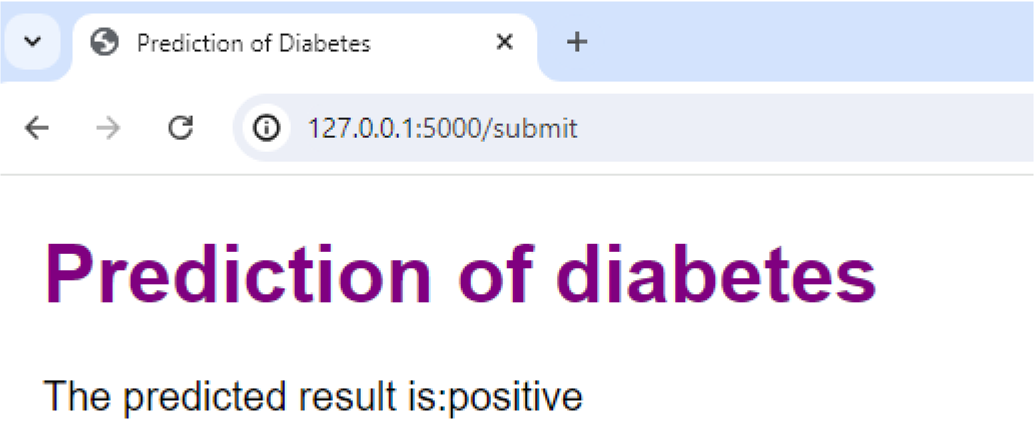
Fig 6.4 after clicking on the link http://127.0.0.1:5000 the above page will be displayed .



Diabetes Prediction

Age: 48
Gender: Female
Polyuria: Yes
Polydipsia: Yes
sudden_weight_loss: Yes
weakness: Yes
Polyphagia: Yes
Genital_thrush: Yes
visual_blurring: Yes
Itching: Yes
Irritability: Yes
delayed_healing: Yes
partial_paresis: Yes
muscle_stiffness: Yes
Alopecia: Yes
Obesity: Yes

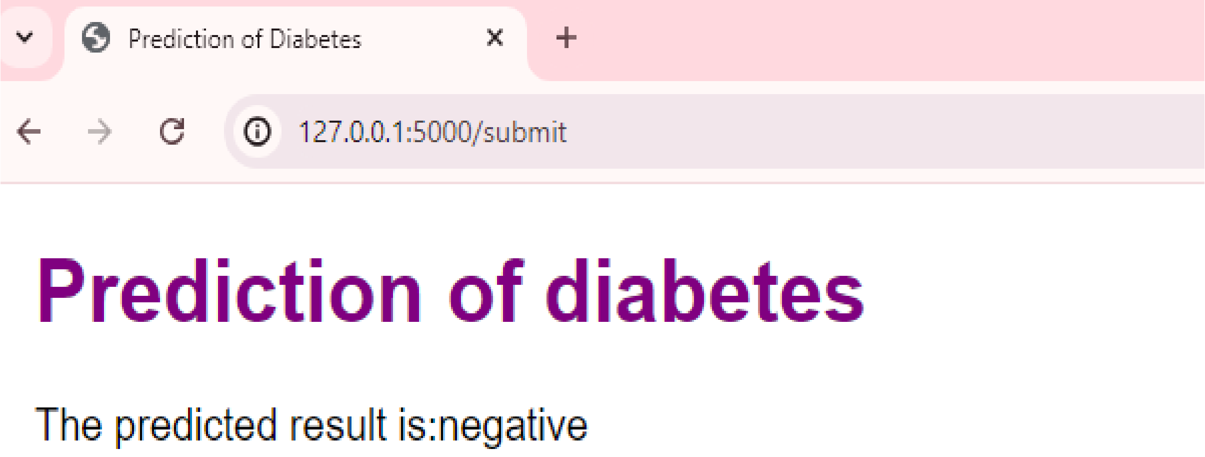
Fig 6. 5 user will update the symptoms in the above page and click on the predict button



Prediction of diabetes

The predicted result is:positive

Fig 6. 6 Based on the sysmptoms Predicted Result



Prediction of diabetes

The predicted result is:negative

Fig 6.7 Based on the symptoms the predict result

Model Accuarcy Comparision for Existed

In this below graph represents the model accuracy camparsion for the SVM and ANN

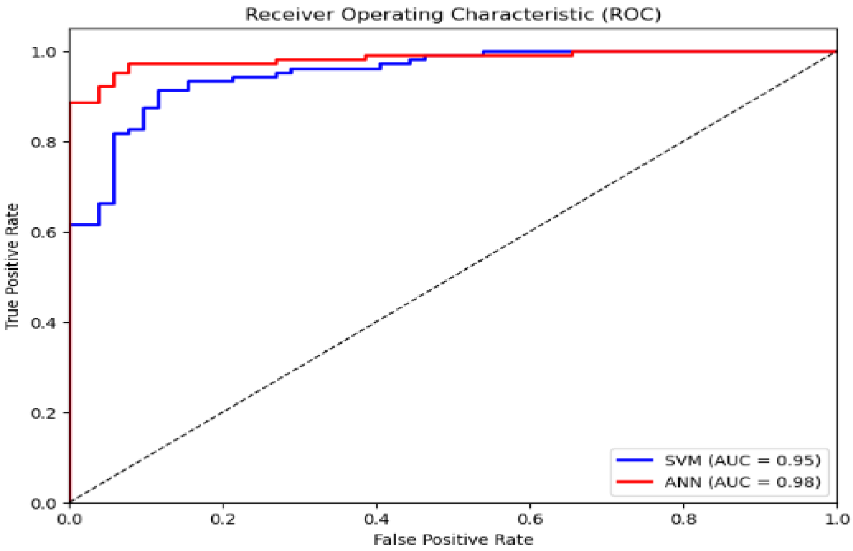
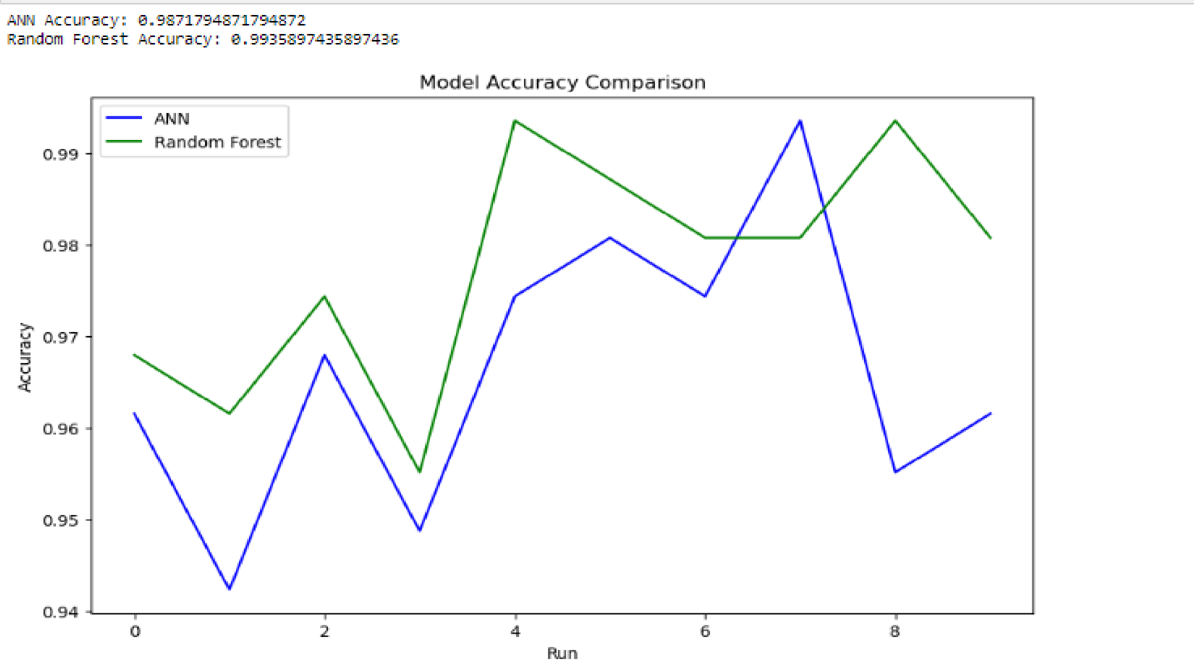


Fig 6.18 (a) Model Accuarcy comparision of Existed

Model Accuracy Comparision for Proposed

In this below graph represents the model accuracy camparsion for the ANN and RF



```
ANN Accuracy: 0.9871794871794872
Random Forest Accuracy: 0.9935897435897436
```

Fig 6.18 (b) Model Accuracy comparision of Proposed

CHAPTER – 7

CONCLUSION AND FUTURESCOPE

CHAPTER – 7

CONCLUSION AND FUTURESCOPE

7.1 CONCLUSION

Despite the use of various models for diabetes prediction, researchers have consistently prioritized improving the accuracy of these models. To address this concern, a new model is necessary to enhance prediction accuracy. This research introduces a machine learning fusion to predict the diabetes . By integrating two widely used machine learning techniques with voting classifier , the proposed decision system achieves an accuracy of 98%, surpassing other existing systems. This improved diagnosis model has the potential to save many lives and help control the death rate associated with diabetes by enabling early diagnosis and preventative measures.

In this paper, we proposed a web page for the successful prediction of diabetes. Our approach for diabetes disease prediction using a machine learning algorithm demonstrates significant potential in accurately detecting various medical data.It takes the real time symptoms from the user and predict the diabetes postive or negative.

7.2 FUTURE SCOPE

The above project implemented web page for the successful prediction of diabetes. Our approach for diabetes disease prediction using a machine learning algorithm demonstrates significant potential in accurately detecting various medical data. In the future, our focus will be on using a deep learning model and preparing a location-based dataset from real medical data for the successful prediction of diabetes.

REFERENCES

REFERENCES

[1] M. A. Sarwar, N. Kamal, W. Hamid, and M. A. Shah, “Prediction of diabetes using machine learning algorithms in healthcare**,”** in Proc. 24th Int. Conf. Autom. Comput. (ICAC), Sep. 2018, pp. 6–7.

[2] Aiswarya Iyer, S. Jeyalatha and Ronak Sumbaly . “DIAGNOSIS OF DIABETES USING CLASSIFICATION MINING TECHNIQUES” International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.5, No.1, January 2015.

[3] Quan Zou, Kaiyang Qu1 , Yamei Luo, Dehui Yin , Ying Ju and Hua Tang “predicting Diabetes Mellitus With Machine Learning Techniques”**.** Frontiers in genetics published on 6th November 2018.

[4] Prof. Dhomse Kanchan B , Mr. Mahale Kishor M . “Study of Machine Learning Algorithms for Special Disease Prediction using Principal of Component Analysis”. 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication from IEEE Access

[5] Sidong Wei , Xuejiao Zhao , Chunyan Miao **.** “A Comprehensive Exploration to the Machine Learning Techniques for Diabetes Identification” from IEEE Xplore.

[6] Samrat Kumar Dey, Ashraf Hossain , Md. Mahbubur Rahman **.** “Implementation of a Web Application to Predict DiabetesDisease: An Approach Using Machine Learning Algorithm”. 21st International Conference of Computer and Information Technology (ICCIT) 21-23 December, 2018, IEEE Access

[7] J.Lysa Eben , R.Jayasudha , S. Ramya , S.Kaliappan, Shobha Aswal , Khalid Ali Salem Al-Salehi .“Diabetes Prediction Model for Better Clarification by using Machine Learning”. Proceedings of the International Conference on Inventive Computation Technologies (ICICT 2023) IEEE Access

[8] Md. Faisal Faruque, Asaduzzaman , Iqbal H. Sarker . "Performance Analysis of Machine Learning Techniques to Predict Diabetes Mellitus". 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), 7-9 February2019, IEEE Access

[9] Ayman Mir , Sudhir N. Dhage . "Diabetes Disease Prediction using Machine Learning on Big Data of Healthcare". 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA),IEEE Access

[10] Shameem Hasan . "Prediction of Diabetes Based on Artificial Intelligence Technique". International Research Journal of Engineering and Technology (IRJET) Volume: 05 Issue: 11 | Nov 2018.

[11] Kezban Alpan, Galip Savasilgi **.** "Classification of Diabetes Dataset with Data Mining Techniques by Using WEKA Approach" From IEEE Accesss

[12] Saloni Kumari , Deepika Kumar , Mamta Mittal . "An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier". International Journal of Cognitive Computing in Engineering.

[13] Soumayadeep Manna, Swagata Maity, Souvik Munshi, Mainak Adhikari **.** "Diabetes Prediction Model Using Cloud Analytics". International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.5, No.1, January 2015.

[14] M. Tech. Scholar Arvind Aada , Prof. Sakshi Tiwari . "Predicting Diabetes in Medical Datasets Using Machine Learning Techniques". International Journal of Scientific Research & Engineering Trends Volume 5, Issue 2, Mar-Apr-2019, ISSN.

[15] Minyechil Alehegn , Rahul Joshi . "Analysis and prediction of diabetes diseases using machine learning algorithm: Ensemble approach". International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 10 | Oct -2017.

[16] G. Pradhan, R. Pradhan, and B. Khandelwal, “A study on various machine learning algorithms used for prediction of diabetes mellitus,” in Soft Computing Techniques and Applications (Advances in Intelligent Systems and Computing), vol. 1248. London, U.K.: Springer, 2021.

[17] S. Saru and S. Subashree. Analysis and Prediction of Diabetes Using Machine Learning. Accessed: Oct. 22, 2022.

[18] P. Sonar and K. JayaMalini, “Diabetes prediction using different machine learning approaches,” in Proc. 3rd Int. Conf. Comput. Methodologies Commun. (ICCMC), Mar. 2019, pp. 367–371,

[19] B. Jain, N. Ranawat, P. Chittora, P. Chakrabarti, and S. Poddar, “A machine learning perspective: To analyze diabetes,” Mater. Today: Proc., pp. 1–5, Feb. 2021,

[20] N. B. Padmavathi, “Comparative study of kernel SVM and ANN classifiers for brain neoplasm classification,” in Proc. Int. Conf. Intell. Comput., Instrum. Control Technol. (ICICICT), Jul. 2017, pp. 469–473,

[21] R. Motka, V. Parmarl, B. Kumar and A. R. Verma, "Diabetes mellitus forecast using different data mining techniques," 2013 4th International Conference on Computer and Communication Technology (ICCCT),

[22] V. Vijayan, and A. Ravikumar, “Study of Data Mining algorithms for Prediction and Diagnosis of Diabetes Mellitus,” International Journal of Computer Application, Vol 94, pp .12-16, June 2014

[23] C. Kalaiselvi and G. M. Nasira, "A New Approach for Diagnosis of Diabetes and Prediction of Cancer Using ANFIS," 2014 World Congress on Computing and Communication Technologies, Trichirappalli, 2014, pp. 188-190.

[24] Kavakiotis, Ioannis, Olga Tsave, AthanasiosSalifoglou, NicosMaglaveras, IoannisVlahavas, and IoannaChouvarda. "Machine learning and data mining methods in diabetes research." Computational and structural biotechnology journal (2017).

[25] Zheng, Tao et al. "A machine learning-based framework to identify type 2 diabetes through electronic health records." International journal of medical informatics 97 (2017): 120- 127.