MACHINE LEARNING APPROACHES FOR DETECTING AND PREDICTING DIABETES-RELATED COMORBIDITIES

DONTALA KIRANKUMAR [1], THOTA PARAMESH [2], ANGATI NAVYA[3], VANGAPANDU VENKATA KALYANI[4]

[1] Assistant professor, Department of Computer Science & Engineering-AI&ML,DS, AVANTHI INSTITUTE OF ENGINEERING& TECHNOLOGY, Cherukupally (V), Near Tagarapuvalsa Bridge,Vizianagaram (Dist)-531162, Andhra Pradesh
Email Id: kirankumartech18@gmail.com

[2] Assistant professor, Department of Computer Science & Engineering-AI&ML,DS, AVANTHI INSTITUTE OF ENGINEERING& TECHNOLOGY, Cherukupally (V), Near Tagarapuvalsa Bridge,Vizianagaram (Dist)-531162, Andhra Pradesh
Email Id: paramesh.thota503@gmail.com

[3]Assistant professor, Department of Computer Science & Engineering, RAGHU ENGINEERING COLLEGE, Dakamarri(v) , Bheemuni Patnam Mandal, Visakhapatnam district-531162, Andhra Pradesh
Email Id: angatinavya1987@gmail.com

[4]Assistant professor, Department of Computer Science & Engineering, RAGHU ENGINEERING COLLEGE, Dakamarri(v) , Bheemuni Patnam Mandal, Visakhapatnam district-531162, Andhra Pradesh
Email Id:kalyani1.mith@gmail.com

ABSTRACT

Diabetes is a chronic metabolic disorder characterized by persistently elevated blood glucose levels. Over time, this condition often leads to several comorbidities, including neuropathy, cardiovascular disease, renal complications, and kidney failure. Early detection and prediction of these comorbidities are crucial for timely intervention and improved patient outcomes. Traditionally, clinical observations and statistical analyses formed the foundation for managing diabetic comorbidities. While valuable, these approaches were limited in their ability to capture the complexity and diversity of comorbidity patterns. The advent of machine learning (ML) has revolutionized this field by enabling the development of advanced, data-driven predictive models with greater precision and reliability. However, building effective ML models for detecting and predicting diabetes-related comorbidities requires comprehensive patient datasets that may include demographics, medical history, diagnostic test results, and even genetic information. Unlike manual methods, which are often constrained by human subjectivity and the inability to process large volumes of data, ML algorithms can efficiently analyze massive datasets, uncovering hidden correlations and complex patterns that might otherwise remain undetected. Accurate and timely prediction of comorbidities is essential, as these conditions significantly reduce the quality of life in diabetic patients. Machine learning offers the potential to support personalized healthcare by continuously learning from historical data, integrating new information, and refining predictive accuracy over time. By identifying key risk factors and forecasting comorbidity development, ML-driven approaches pave the way for earlier interventions and more effective treatment strategies. In this context, machine learning represents a transformative step toward personalized medicine, offering significant promise in reducing the burden of comorbidities and improving the overall well-being of individuals with diabetes.

KEYWORDS: Machine Learning, Diabetes, Health care

1. INTRODUCTION

The management of diabetic comorbidities has evolved significantly, moving from initial clinical observations and statistical analyses to the revolutionary integration of machine learning techniques. In the early stages, healthcare professionals relied heavily on their expertise and manual analysis of patient data to identify and predict comorbidities. However, this approach was limited by subjectivity, inefficiency in handling large datasets, and an inability to detect complex, non-linear patterns.

The introduction of machine learning marked a paradigm shift, enabling the development of highly sophisticated and accurate predictive models. Unlike traditional methods, machine learning can leverage diverse datasets, including patient demographics, medical histories, diagnostic test results, and even genetic information. The central challenge then became designing algorithms capable of processing such heterogeneous data to predict the likelihood of specific comorbidities in diabetic patients.

Given the profound impact comorbidities have on the quality of life of individuals with diabetes, early detection and timely intervention are critical for improving treatment outcomes. Machine learning has addressed this need by efficiently analyzing massive datasets, uncovering hidden correlations, and providing actionable insights that would otherwise remain undetected through manual methods. This advancement represents a major leap toward personalized medicine, offering proactive and precise healthcare strategies that not only improve patient outcomes but also help reduce the overall burden of comorbidities in diabetes care.

1.2 Research Motivation

Addressing the difficulties of comorbidities with sophisticated machine learning approaches is crucial to improving healthcare outcomes for patients with diabetes. our is the motivating force for our study. Comorbid disorders worsen the health of people with diabetes since the disease is a chronic metabolic disorder. Comorbidity patterns are complex and dynamic, making it difficult for early attempts that relied on clinical observations and statistical analysis to manage them. With the rise of machine learning, there is a chance to build advanced prediction models that can thoroughly analyse various datasets, which might be a great way to circumvent these restrictions. This is driven by the realisation that in order to enhance patient outcomes and facilitate effective management, comorbidity diagnosis must be done accurately and promptly. Due to human error, inefficiency, and the sheer volume of data involved, healthcare providers still rely on time-honoured manual analysis methods. Conversely, machine learning has the ability to revolutionise this area by revealing hidden connections and patterns in patient data, which might result in more tailored healthcare treatments. The ultimate aim is to help develop personalised medicine, where machine learning is crucial in making people's lives better with diabetes by lowering the load of related comorbidities via proactive and accurate healthcare practices.

1.3 Problem Statement

Detecting and predicting comorbidities in people with diabetes, a chronic metabolic illness defined by increased blood sugar levels, is the problem that this study aims to solve. Statistical analysis and clinical observations were the first approaches used for this, but they couldn't cope with the complexity and variety of comorbidity patterns. More accurate prediction models were made possible with the advent of machine learning methods, which signified a paradigm change. The challenge description asks us to build models that can predict which diabetic patients are most likely to acquire certain comorbid diseases using a variety of datasets, such as demographic information, medical records, test results, and maybe even genetic information. Healthcare providers used to rely largely on subjective results, inefficiencies in handling large amounts of data, and missed complex patterns that machine learning models could detect and predict when it came to diabetes-related comorbidities. A break from conventional wisdom is required to meet the critical need for rapid and precise comorbidity diagnosis. By studying massive datasets, finding predictive traits, and increasing its forecast accuracy over time with the help of both historical and fresh data, machine learning offers a potential answer. With any luck, we'll be able to lessen the toll that diabetes and its complications have on people's lives by making healthcare treatments more efficient.

1.4 Application

Potentially game-changing for healthcare systems and patients alike are the results of this study's applicability to the use of machine learning for the diagnosis and prognosis of comorbidities in diabetic patients. Machine learning algorithms may assess various patient information, such as demographics, medical history, and laboratory findings, to determine the probability of certain comorbidities. This has important implications for personalised healthcare treatments. Better, more focused therapies are possible now that healthcare practitioners may personalise interventions according to patients' risk profiles. The results of this study may also help in the creation of early warning systems, which can help in the prevention or reduction of the effects of new comorbidities by allowing for prompt treatments. Machine learning's use in diabetes treatment has the potential to influence public health policy, resource distribution, and preventative measures. In addition, by using predictive models, patients may be better educated and empowered to take an active role in treating their diabetes and any comorbidities. Patient care, healthcare efficiency, and overall human well-being in the face of chronic metabolic illnesses may all be greatly improved with the help of machine learning technologies in this context, thanks to their flexibility and accuracy.

2. LITERATURE SURVEY

Multiple classifiers, including SVM, J48, K-Nearest Neighbours (KNN), and Random Forest, were suggested by Kandhasamy and Balamurali [1]. A dataset obtained from the UCI repository was used for the classification. We compared the classifiers' outputs according to their sensitivity, specificity, and accuracy scores. Using 5-fold cross validation, classification was performed in two scenarios: one with and one without preprocessing the dataset. While the authors did note that the dataset was pre-processed to eliminate noise, they did not elaborate on the specific steps used to do so. The decision tree J48 classifier was found to have the greatest accuracy rate of 73.82% when data was not pre-processed, whereas Random Forest and KNN (k = 1) classifiers had a 100% accuracy rate when data was pre-processed.

In their presentation, Yuvaraj and Sripreethaa [2] demonstrated a diabetes prediction application that used three distinct machine learning algorithms: Random Forest, Decision Tree, and Naïve Bayes. Preprocessing was done on the Pima Indian Diabetes dataset (PID) before it was utilised. The authors didn't go into detail on the data pre-processing, but they did talk about the feature selection approach they employed, the Information Gain method, to get the right features. Out of thirteen, they relied on only eight primary qualities. Also, 70% of the dataset was put aside for training purposes, while 30% was kept for testing. A total of 94% of the time, the random forest method proved to be the most accurate.

In addition, Tafa [3] suggested a novel model that combines SVM and Naïve Bayes for diabetes prediction. Datasets acquired from three separate sites in Kosovo were used to assess the model. Out of 402 individuals included in the sample, 80 had type 2 diabetes. There are a total of eight characteristics. The regular diet, physical activity, and family history of diabetes are some of the features that have not been studied previously but are used in this research. The data's pre-processing status was not disclosed by the writers. They used a 50/50 split between the dataset's training and testing sets to conduct the validation test. Thanks to the suggested combination methods, the prediction accuracy is now 97.6%. The performance of SVM (95.52%) and Naïve Bayes (94.52%) were compared with this value.

Also, Deepti and Dilip [4] identified diabetes using Decision Tree, Support Vector Machine, and Naive Bayes classifiers. The goal was to find the most accurate classifier. This research made use of the Pima Indian dataset. Using 10-folds cross-validation, the dataset is divided. Data preparation was not addressed by the writers. Accuracy, precision, recall, and the F-measure were the metrics used to assess the performance. Naive Bayes had the best accuracy at 76.30 percent.

Six distinct classifiers were suggested by Mercaldo [5]. J48, Random Forest, Bayes Net, Rip, HoeffdingTree, and Multilayer Perceptron are the classifiers. This research also made use of the Pima Indian dataset. Although no preprocessing step was specified, the authors used two methods, BestFirst and GreedyStepwise, to identify discriminating features that improved classification accuracy. We have narrowed the criteria down to four: age, diabetic pedigree function, body mass index, and plasma glucose concentration. The dataset is subjected to a 10-fold cross-validation. We compared the classifiers using three metrics: recall, precision, and the F-Measure. Using the Hoeffding Tree technique, the results showed an accuracy value of 0.757, a recall value of 0.762, and an F-measure of 0.759. When compared to the previous performances, this one is at the top.

Negi and Jaiswal [5] sought to use the SVM for diabetes prediction, similar to prior research. We utilised a composite dataset that included both the Pima Indians and Diabetes 130-US studies. Since previous studies often relied on a single dataset, this one set out to prove that the findings could be trusted. There are a total of 102,538 samples in the dataset, with 64,419 being positive and 38,115 being negative. The collection also includes 49 qualities. In this research, the authors choose not to reveal which characteristics were used. Before normalisation between 0 and 1, the dataset undergoes pre-processing that involves replacing missing values and out-of-range data with zero, converting non-numerical values to numerical values, and ultimately, the data is cleaned up. Before the SVM model was used, several feature selection strategies were employed. The LIBSVM package's Fselect script picked four characteristics, but the Weka Tool's Wrapper and Ranker methods picked nine and twenty attributes, respectively. The authors used a 10-fold cross validation approach for the validation procedure. The 72% accuracy rate of the diabetes prediction suggests that a pooled dataset may provide more accurate results.

Additionally, a Multilayer Feed-Forward Neural Network was used by Olaniyi and Adnan [6]. The algorithm was trained using the back-propagation approach. Increasing the precision of diabetes prediction was the primary objective. This study made use of the Pima Indian Diabetes Registry. For numerical stability, the authors normalised the dataset before running the classification. The process included converting all dataset values to integers between 0 and 1 by dividing each sample attribute by its associated amplitude. The next step is to split the dataset in half, with half going into a training set and the other half into a testing set. An very high accuracy rate of 82% was achieved.

In order to foretell the onset of diabetes, Soltani and Jafarian [7] used a Probabilistic Neural Network (PNN). The Pima Indian dataset was subjected to the algorithm's application. The writers did not use any pre-processing method. There is a 90% training set and a 10% testing set inside the dataset. With 89.56% accuracy on training data and 81.49% on testing data, the suggested method was successful.

For numerical stability, Rakshit [8] pre-processed the dataset by normalising the values of all sample characteristics using the mean and standard deviation of each variable. Furthermore, the pertinent characteristics were recovered by means of the correlation. Nevertheless, these biassed characteristics were not mentioned by the writers. The dataset was divided into two parts: one with 314 samples for training and another with 78 samples for testing. When compared to other accuracies acquired from earlier research, this model's outcome earned the best accuracy at 83.3%.

Levenberg Marquardt (LM), Bayesian Regulation (BR), and Scaled Conjugate Gradient (SCG) were three supervised learning techniques that Mamuda and Sathasivam [9] used. The Pima Indian dataset was used in this research to assess performance. The dataset has 768 samples and eight variables. The 10-fold cross validation was used to divide the data into training and testing sets for the validation investigation. Since the Mean Squared Error (MSE) for the validation set was 0.00025091, the authors concluded that Levenberg Marquardt (LM) performed the best.

Logistic regression was suggested by Mohebbi [10] as a foundational layer for both conventional and multilayer perceptron neural networks. Using a dataset of continuous glucose monitoring (CGM) signals, the goal was to identify diabetes individuals. There are a total of 97,200 simulated CGM days in the dataset, which is comprised of nine individuals. Each patient has 10,800 days of CGM data. In this investigation, the characteristics that were used were not revealed. This dataset was divided into three parts using the leave-one-patient-out cross-validation technique: training, validation, and testing. Six patients were really chosen for training and validation, while three were chosen for testing, by the authors. With an accuracy of 77.5%, the CNN obtained the best result.

An unsupervised Deep Neural Network architecture, Deep Patient, was suggested by Miotto [11]. A database of electronic health records including information on 704,857 patients was used by the framework. Although the authors did note that the dataset may be used for illness prediction, they did not elaborate on the elements that made up the dataset. The scientists divided the data into three groups: 5,000 patients for validation, 76,217 for testing, and the remaining patients for training. The Area Under the Curve (AUC) was 0.91, indicating good accuracy. If you want your predictions to turn out better, the authors say you should pre-process the dataset. They recommended using PCA to get the important properties before running the DL.

Pham [12] used a dataset obtained by hand from a rural hospital in Australia and applied three distinct DL methods on it. Of the 12,000 samples (patients) included in the dataset, 55.5% are men. In order to clean and decrease the samples to 7191 patients, some pre-processing procedures were used, however they are not stated in their paper. Half of the dataset was used for training, one half for validation, and one half for testing purposes during validation. Markov, Plain RNN, and Long Short-Term Memory (LTSM) were the approaches used. In order to evaluate how well each method worked, we looked at their accuracy values. The LTSM yielded the highest accuracy rate of 59.6 percent.

To forecast the two forms of diabetes, Ramesh [13] used a Recurrent Neural Network (RNN). They used the 768-sample, 8-attribute Pima Indian dataset. Based on the research titled "Glucose, BMI, Age, Pregnancies, Diabetes Pedigree Function, Blood Pressure, Skin Thickness and Insulin," the features are ranked in order of the most important ones. In order to ensure the study's validity, the dataset was divided into two parts: the training set and the testing set. Predicting type 2 diabetes was 81% accurate, whereas type 1 diabetes was 78% accurate.

Lekha and Suchetha [14] used one-dimensional modified CNN in their analysis of breath signals for diabetes prediction, among previous investigations. Eleven healthy individuals, nine type 2 diabetics, and five type 1 diabetics made up the breath signal dataset they gathered. This dataset's characteristics are shown. The dataset was not pre-processed. Leave-One Out Cross Validation was used by the writers for the validation procedure. The Receiver Operating Characteristics (ROC) curve, which achieved 0.96, was used to assess the performance.

3. PROPOSED SYSTEM

The Python source code demonstrates machine learning research aimed at identifying and predicting diabetic comorbidities using multiple methods. The implemented procedures are summarized as follows:

- Dataset Upload and Display: At the beginning, the user is prompted to upload a dataset containing information related to diabetic comorbidities. Through the Tkinter-based GUI, a file dialog allows the user to select the dataset file. Once loaded, the file path is displayed, along with a preview of the first few rows of the dataset.
- Dataset Preparation: After uploading, the user initiates dataset preparation. Missing values are replaced with zeros, and the data is converted into NumPy arrays for further processing. The dataset is normalized using standard scaling techniques, and features are separated from labels.
- Data Splitting: The normalized dataset is divided into training and testing subsets using the `train_test_split` function from Scikit-Learn. Dataset statistics, including the total number of records and the distribution between training and testing sets, are also displayed.
- Naive Bayes Implementation: A Gaussian Naive Bayes model is trained on the training set and used for disease prediction on the test set. Performance metrics such as accuracy, precision, recall, and F1-score are calculated, along with a confusion matrix for visualization.
- Artificial Neural Network (ANN) Model: An ANN-based model is also implemented using Keras. A multi-layer neural network is constructed and trained on the training set, with model weights saved for future use. If pre-trained weights exist, they are automatically loaded. The test set is used for evaluation, and performance measures are generated in the same manner as for the Naive Bayes model.
- Performance Comparison: The performance of both Naive Bayes and ANN models is compared. Metrics such as accuracy, precision, recall, and F1-score are displayed, and a comparison graph is generated to highlight the relative effectiveness of the two approaches.
- Prediction on New Data: The system also allows users to upload new test data for prediction. The ANN model generates predictions for each entry, and the GUI presents the results, including the uploaded test data alongside the predicted outcomes.

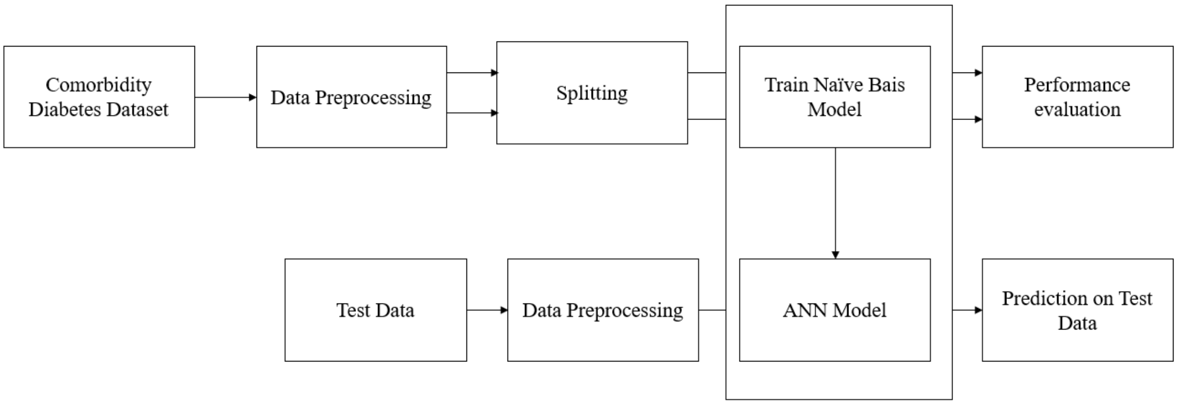


Figure 3.1: Block Diagram of Proposed Model.

3.2 Dataset Splitting

We split our dataset in half, creating a training set and a test set, as part of the pre-processing phase of machine learning data. Because this improves our machine learning model's performance, it is an important part of data pre-processing. Say we've trained our machine learning model on one dataset and then tested it on another. That will make it harder for our model to deduce model-to-model relationships.

Even if our model is very accurate during training, its performance will suffer when we feed it a different dataset. Making a machine learning model that does well on both the training and test sets is, therefore, our goal. This is where various datasets may be described:

Known as the "training set," this is a subset of the dataset used to train the machine learning model.

A subset of the dataset used to evaluate the machine learning model's output predictions.

3.3 ANN Classifier

The Perceptron's original intent was as an image recognition machine, but now it's more well known as an algorithm. Its ability to observe, perceive, and identify visual content is whence it derives its name. We are all familiar with the "phenomenal world" where light, sound, temperature, etc., come from, and the idea of a machine that can conceptualise these inputs directly from the physical environment, without requiring a human agent to digest and code the information, has been the focus of much interest. The neurone was the fundamental building block of Rosenblatt's perceptron machine. A neuron's cell, like in earlier models, takes in a sequence of inputs and weights in pairs. The key modification to Rosenblatt's model is the incorporation of a weighted aggregate of inputs; neurones are programmed to fire and provide an output only when this weighted total over a certain threshold.

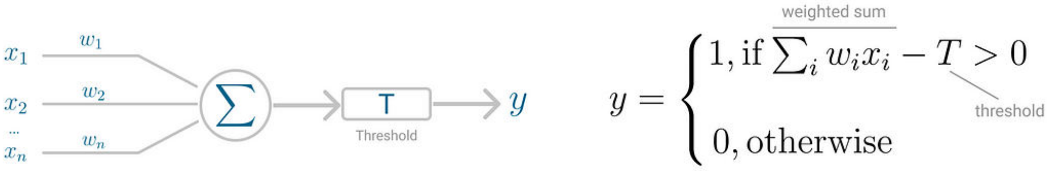

x_1
w_1
x_2
w_2
...
x_n
w_n
$\sum$
T
Threshold
y
weighted sum
$y = \begin{cases} 1, \text{if } \sum_i w_i x_i - T > 0 \\ 0, \text{otherwise} \end{cases}$
threshold

Fig. 3.2: Perceptron neuron model (left) and threshold logic (right).

Threshold T represents the activation function. If the weighted sum of the inputs is greater than zero the neuron outputs the value 1, otherwise the output value is zero.

Perceptron for Binary Classification

The activation function controls the perceptron's discrete output, which defines a linear decision boundary and allows it to be utilised as a binary classification model.

By minimising the distance between the decision border and misclassified locations, it discovers the separating hyperplane. As shown below, the perceptron loss function:

$$\underbrace{D(w,c)}_{\text{distance}} = -\sum_{\underset{\text{misclassified observations}}{i\in \mathrm{M}}} \overset{\text{output}}{y_i}\left(x_i w_i + c\right)$$

Perceptron employs stochastic gradient descent (SGD) as its optimisation function to reduce this gap to a minimum. We guarantee that SGD will converge in a limited number of steps if the data is linearly separable. Last but not least, Perceptron requires the activation function, which is responsible for deciding whether or not a neurone will fire. You can see it makes sense by looking at the form of the sigmoid function, which was employed by the first Perceptron models. The sigmoid function encodes a non-linear function by mapping any actual input to a value of either 0 or 1. Even when fed negative integers, the neurone can only ever provide a positive or negative number as an output.

Although, the Rectified Linear Unit (ReLU) is the neuron's activation function in the vast majority of Deep Learning algorithms and publications published in the last ten years. Because it is scale-invariant—that is, its properties are unaffected by the size of the input—and because it enables better optimisation using SGD, ReLU became increasingly popular. In response to inputs, the neurone randomly selects a starting set of weights. The output value is determined by ReLU, the activation function, after they are merged in a weighted sum.

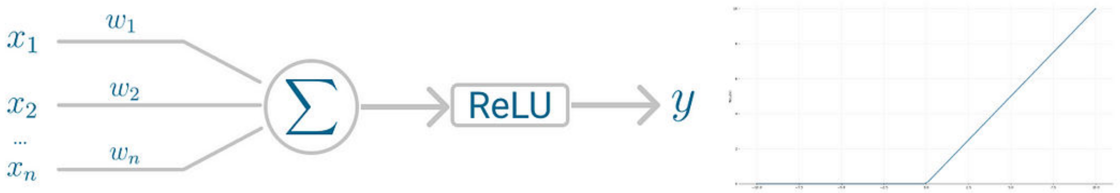


Fig. 3.3: Perceptron neuron model (left) and activation function (right).

The perceptron learns, or finds, the weights that minimise the distance between the decision border and the misclassified points using SGD. A linear hyperplane divides the dataset in half when SGD converges. Despite claims that the Perceptron could model any logic or circuit, the most common complaint was that it couldn't simulate the XOR gate, also known as exclusive OR, which simply returns 1 when the inputs are different. This confirmed itself almost ten years later, and it emphasises how the Perceptron, which only has one neurone, is ill-suited for non-linear data.

3.3.2 ANN

This constraint was the inspiration for the development of the ANN. A non-linear mapping from inputs to outputs characterises this neural network. Neural networks (ANNs) consist of several stacked neurones in a hidden layer or layers, as well as an input and output layer. Additionally, neurones in an ANN are free to employ whatever activation function they wish, unlike in a Perceptron, which requires activation functions like ReLU or sigmoid that enforce a threshold. Similar to the Perceptron, ANN is a feedforward algorithm as it uses a weighted sum of inputs and starting weights to apply the activation function. The key distinction, however, is the layer-by-layer propagation of each linear combination. The outcome of each layer's calculation, or its internal representation of the data, is passed on to the next layer. This continues on to the output layer after passing through all of the hidden levels. The algorithm would be unable to learn the weights that minimise the cost function if it only calculated the weighted sums in each neurone and communicated the results to the output layer. There would be no real learning if the algorithm simply calculated one iteration. Here is where the concept of backpropagation is used.

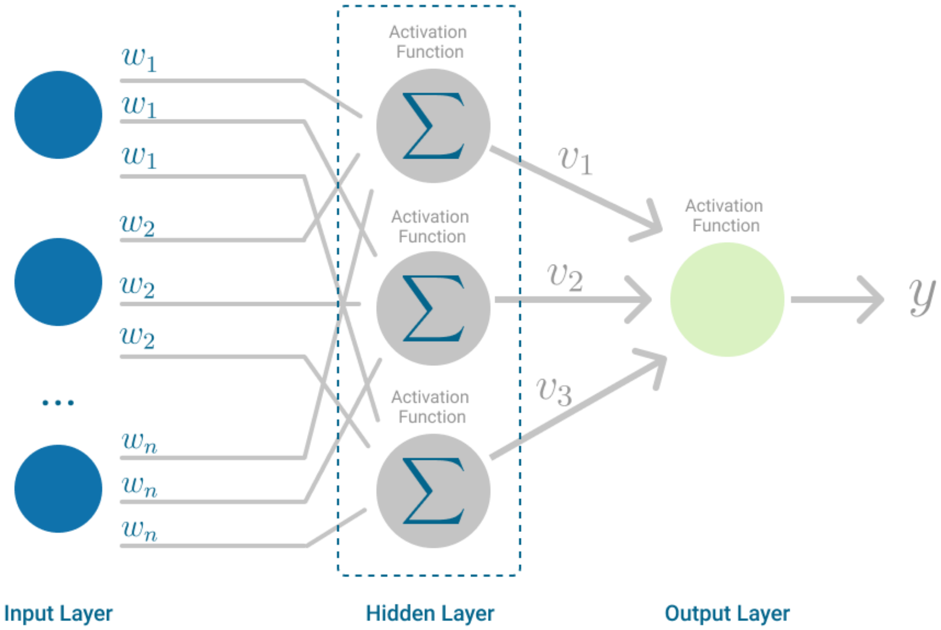


Fig. 3.4: Architecture of ANN.

Backpropagation: The artificial neural network (ANN) learns to minimise the cost function by repeatedly adjusting the network's weights via backpropagation. In order for backpropagation to function correctly, one strict condition must be met. The differentiability of the threshold function (ReLU) and the function that integrates neurone inputs and weights (weighted sum, for example) are prerequisites for their use. The optimisation function usually employed in ANN is Gradient Descent, hence these functions must have a limited derivative. After passing the weighted sums through each layer in an iteration, the gradient of the Mean Squared Error is calculated for every pair of inputs and outputs. Then, to send it back, we update the first hidden layer's weights with the gradient's value. In this way, the weights are sent all the way back to the neural network's initialisation point. A specific version of Gradient Descent is described here:

$$\underbrace{\Delta_w(t)}_{\text{Gradient Current Iteration}} = -\overset{\text{Bias}}{\varepsilon}\underbrace{\frac{\overset{\text{Error}}{dE}}{dw_{(t)}}}_{\text{Weight vector}} + \overset{\text{Learning Rate}}{\alpha}\underbrace{\Delta_{w(t-1)}}_{\text{Gradient Previous Iteration}}$$

The procedure continues until the gradient for each input-output pair converges, which is defined as the point at which the freshly calculated gradient is no longer different from the previous iteration by more than a set convergence threshold.

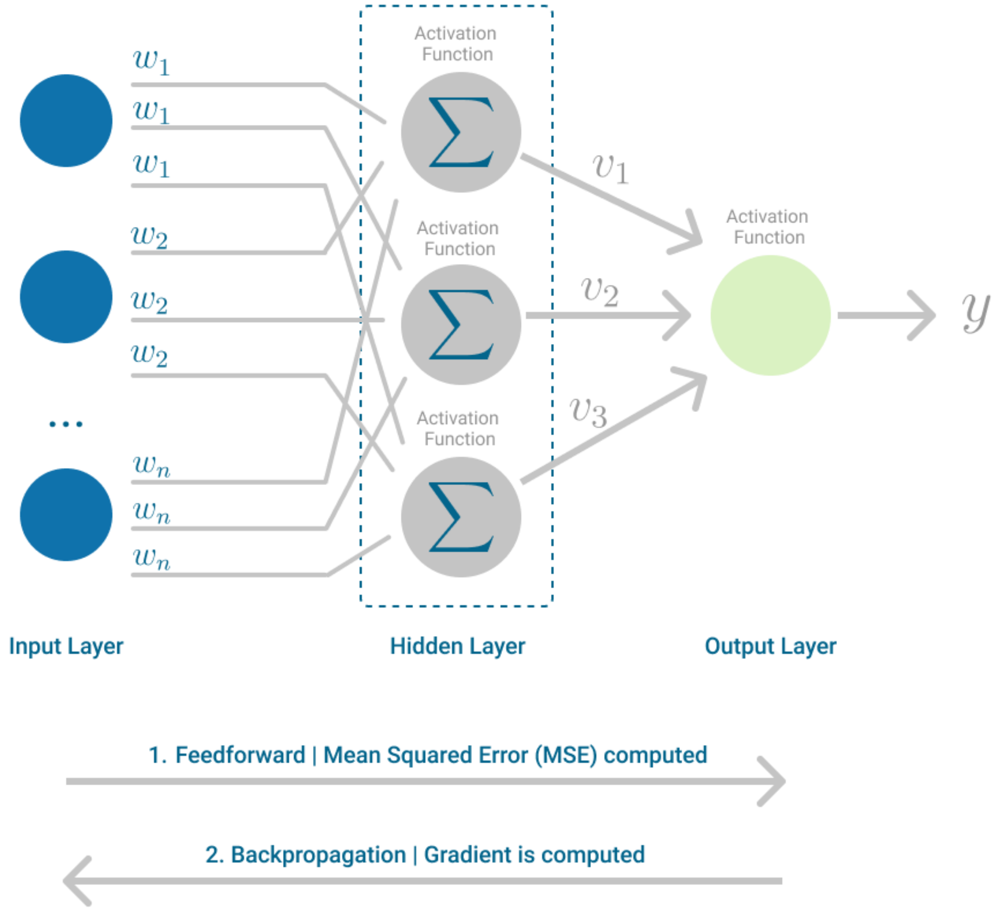
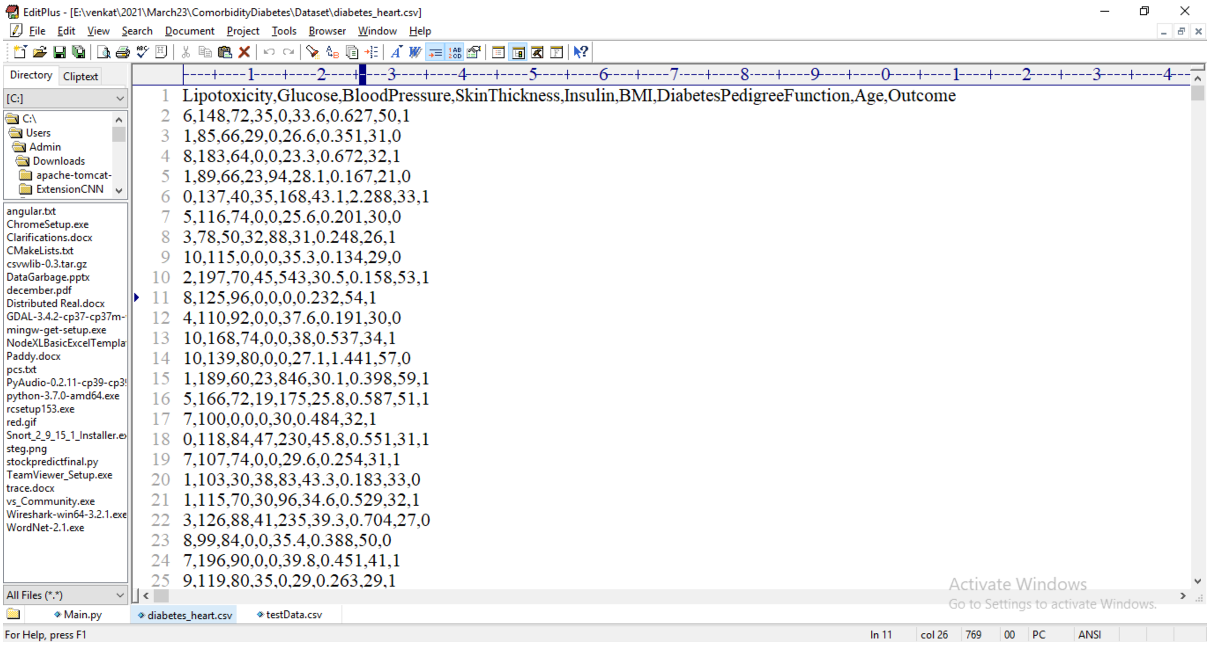


Fig. 3.5: ANN, highlighting the Feedforward and Backpropagation steps.

4. RESULTS



```
Lipotoxicity,Glucose,BloodPressure,SkinThickness,Insulin,BMI,DiabetesPedigreeFunction,Age,Outcome
6,148,72,35,0,33.6,0.627,50,1
1,85,66,29,0,26.6,0.351,31,0
8,183,64,0,0,23.3,0.672,32,1
1,89,66,23,94,28.1,0.167,21,0
0,137,40,35,168,43.1,2.288,33,1
5,116,74,0,0,25.6,0.201,30,0
3,78,50,32,88,31,0.248,26,1
10,115,0,0,0,35.3,0.134,29,0
2,197,70,45,543,30.5,0.158,53,1
8,125,96,0,0,0,0.232,54,1
4,110,92,0,0,37.6,0.191,30,0
10,168,74,0,0,38,0.537,34,1
10,139,80,0,0,27.1,1.441,57,0
1,189,60,23,846,30.1,0.398,59,1
5,166,72,19,175,25.8,0.587,51,1
7,100,0,0,0,30,0.484,32,1
0,118,84,47,230,45.8,0.551,31,1
7,107,74,0,0,29.6,0.254,31,1
1,103,30,38,83,43.3,0.183,33,0
1,115,70,30,96,34.6,0.529,32,1
3,126,88,41,235,39.3,0.704,27,0
8,99,84,0,0,35.4,0.388,50,0
7,196,90,0,0,39.8,0.451,41,1
9,119,80,35,0,29,0.263,29,1
```

Figure 1: Displays the dataset.

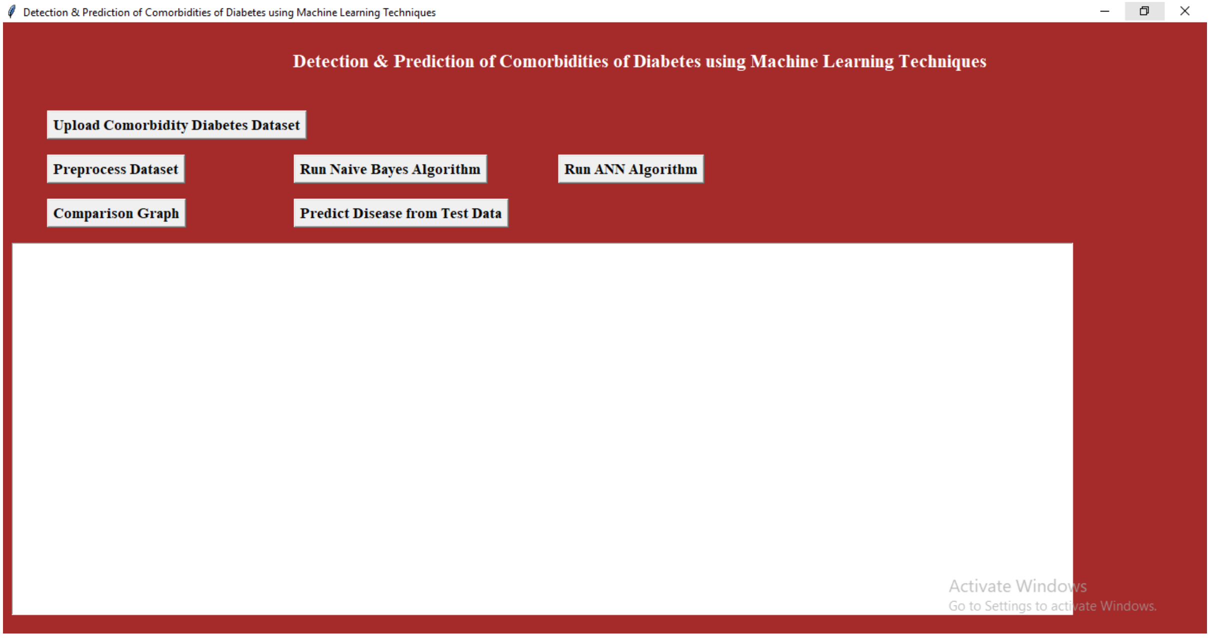
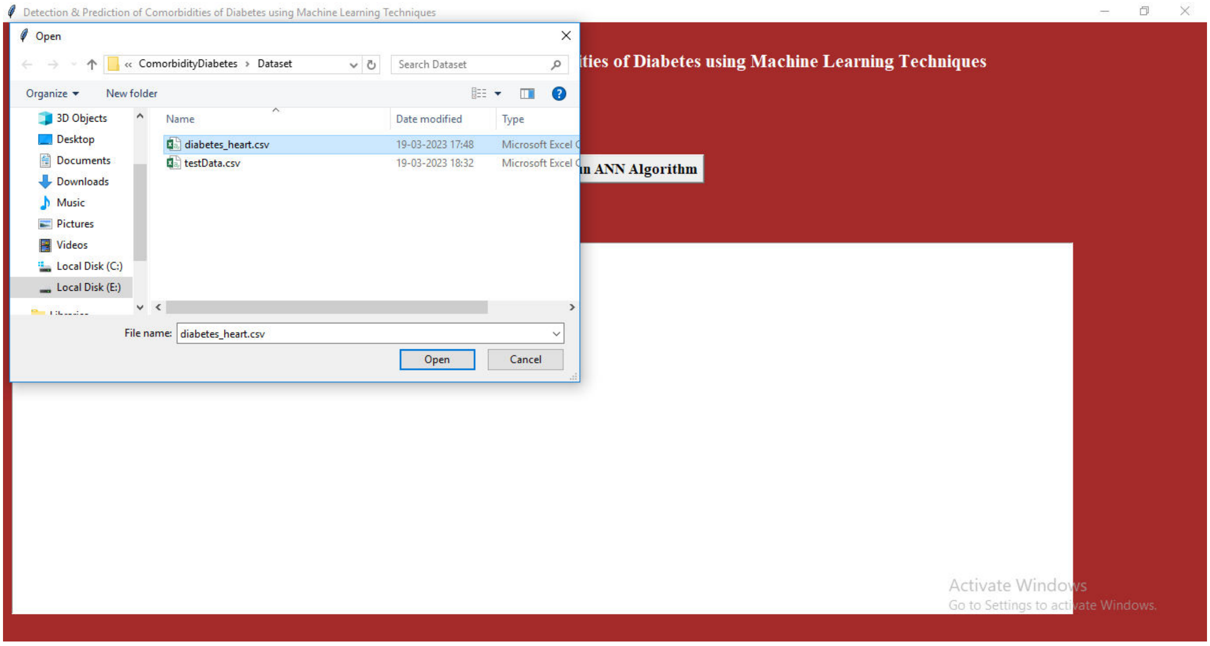

Detection & Prediction of Comorbidities of Diabetes using Machine Learning Techniques
Detection & Prediction of Comorbidities of Diabetes using Machine Learning Techniques
Upload Comorbidity Diabetes Dataset
Preprocess Dataset
Run Naive Bayes Algorithm
Run ANN Algorithm
Comparison Graph
Predict Disease from Test Data
Activate Windows
Go to Settings to activate Windows.

Figure 2: Displays the GUI interface of Comorbidities of Diabetes.


Detection & Prediction of Comorbidities of Diabetes using Machine Learning Techniques
Open
« ComorbidityDiabetes › Dataset
Search Dataset
Organize
New folder
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Name
Date modified
Type
Open
Cancel
ities of Diabetes using Machine Learning Techniques
n ANN Algorithm
Activate Windows
Go to Settings to activate Windows.

diabetes_heart.csv 19-03-2023 17:48 Microsoft Excel C

testData.csv 19-03-2023 18:32 Microsoft Excel C

File name: diabetes_heart.csv

Figure 3: Displays the selection of dataset to upload in the GUI.

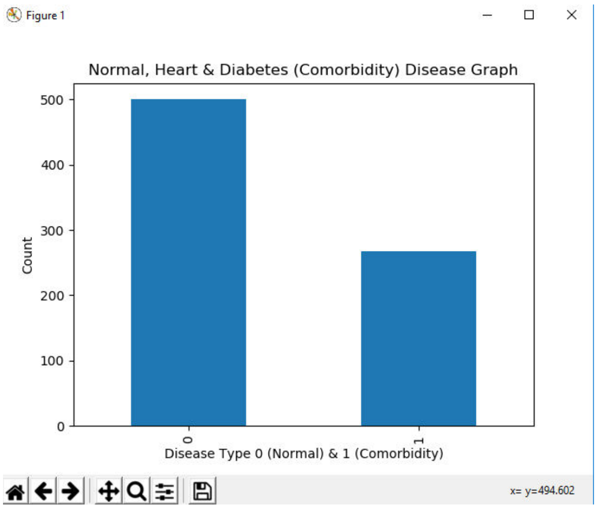
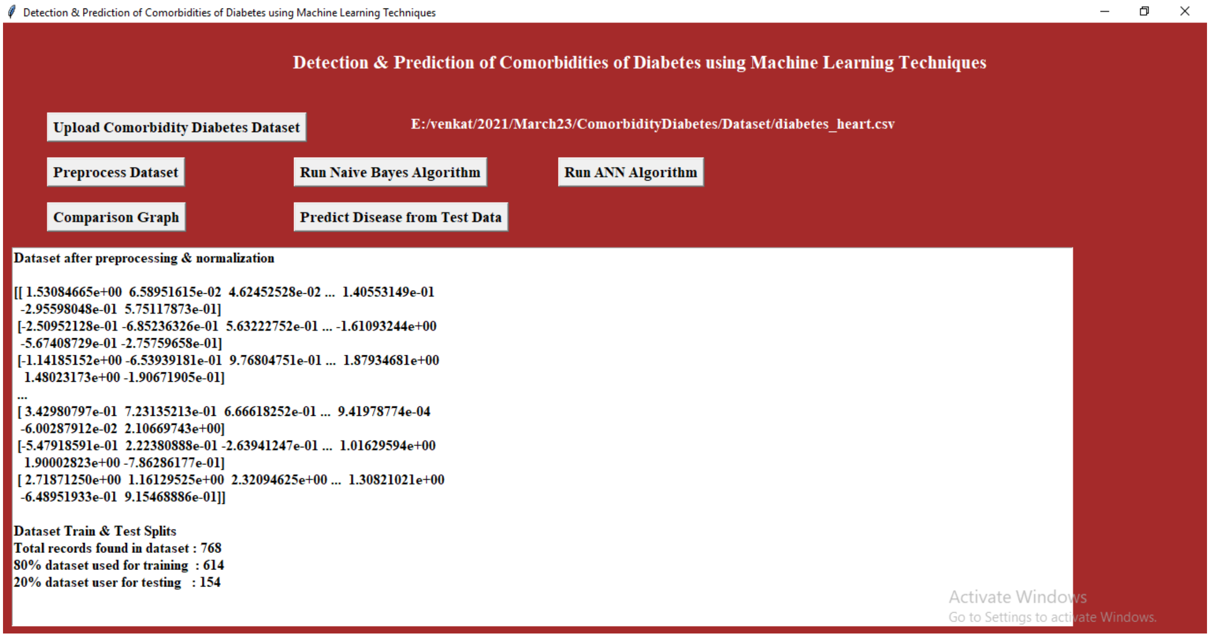


Figure 4: Shows the count plot of each class in the dataset.


Detection & Prediction of Comorbidities of Diabetes using Machine Learning Techniques
Detection & Prediction of Comorbidities of Diabetes using Machine Learning Techniques
Upload Comorbidity Diabetes Dataset
E:/venkat/2021/March23/ComorbidityDiabetes/Dataset/diabetes_heart.csv
Preprocess Dataset
Run Naive Bayes Algorithm
Run ANN Algorithm
Comparison Graph
Predict Disease from Test Data
Activate Windows
Go to Settings to activate Windows.

```
Dataset after preprocessing & normalization

[[ 1.53084665e+00  6.58951615e-02  4.62452528e-02 ...  1.40553149e-01
  -2.95598048e-01  5.75117873e-01]
 [-2.50952128e-01 -6.85236326e-01  5.63222752e-01 ... -1.61093244e+00
  -5.67408729e-01 -2.75759658e-01]
 [-1.14185152e+00 -6.53939181e-01  9.76804751e-01 ...  1.87934681e+00
   1.48023173e+00 -1.90671905e-01]
 ...
 [ 3.42980797e-01  7.23135213e-01  6.66618252e-01 ...  9.41978774e-04
  -6.00287912e-02  2.10669743e+00]
 [-5.47918591e-01  2.22380888e-01 -2.63941247e-01 ...  1.01629594e+00
   1.90002823e+00 -7.86286177e-01]
 [ 2.71871250e+00  1.16129525e+00  2.32094625e+00 ...  1.30821021e+00
  -6.48951933e-01  9.15468886e-01]]

Dataset Train & Test Splits
Total records found in dataset : 768
80% dataset used for training  : 614
20% dataset user for testing   : 154
```

Figure 5: Displays the dataset preprocessing and normalization.

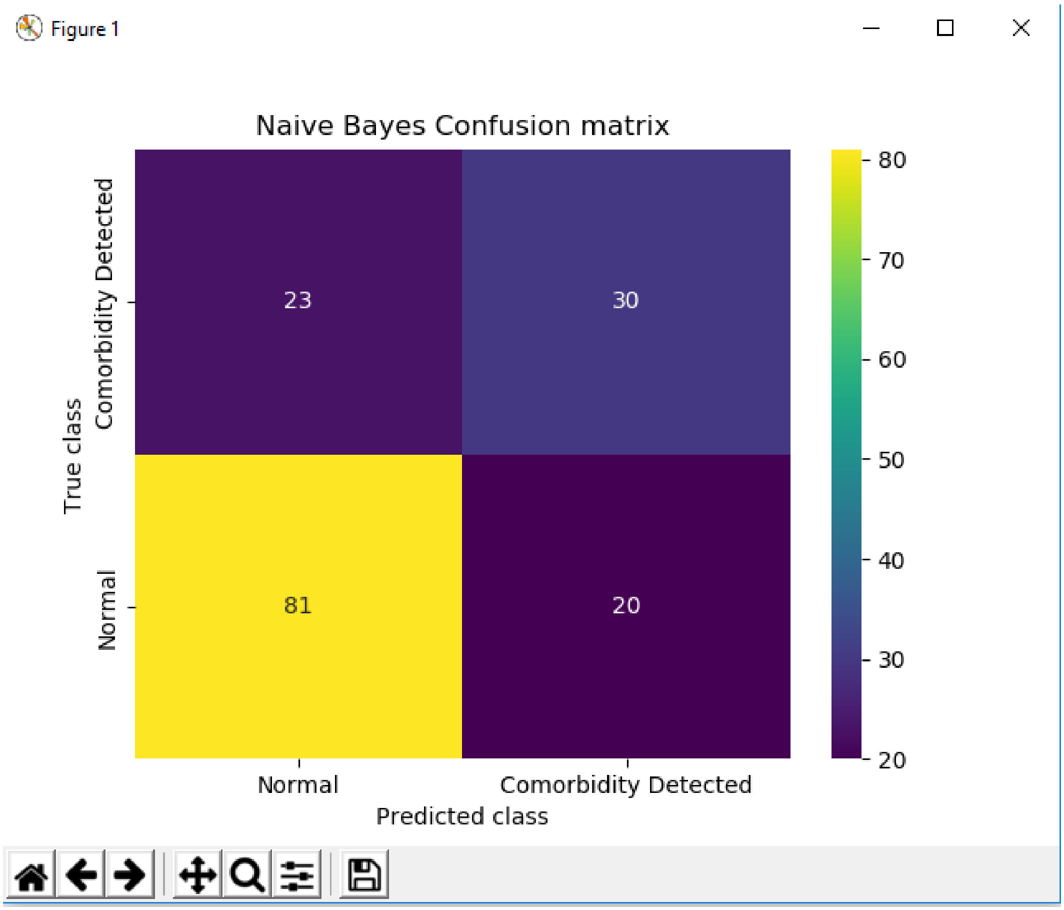


Figure 6: Presents the confusion matrix of Naïve Bayes Model.

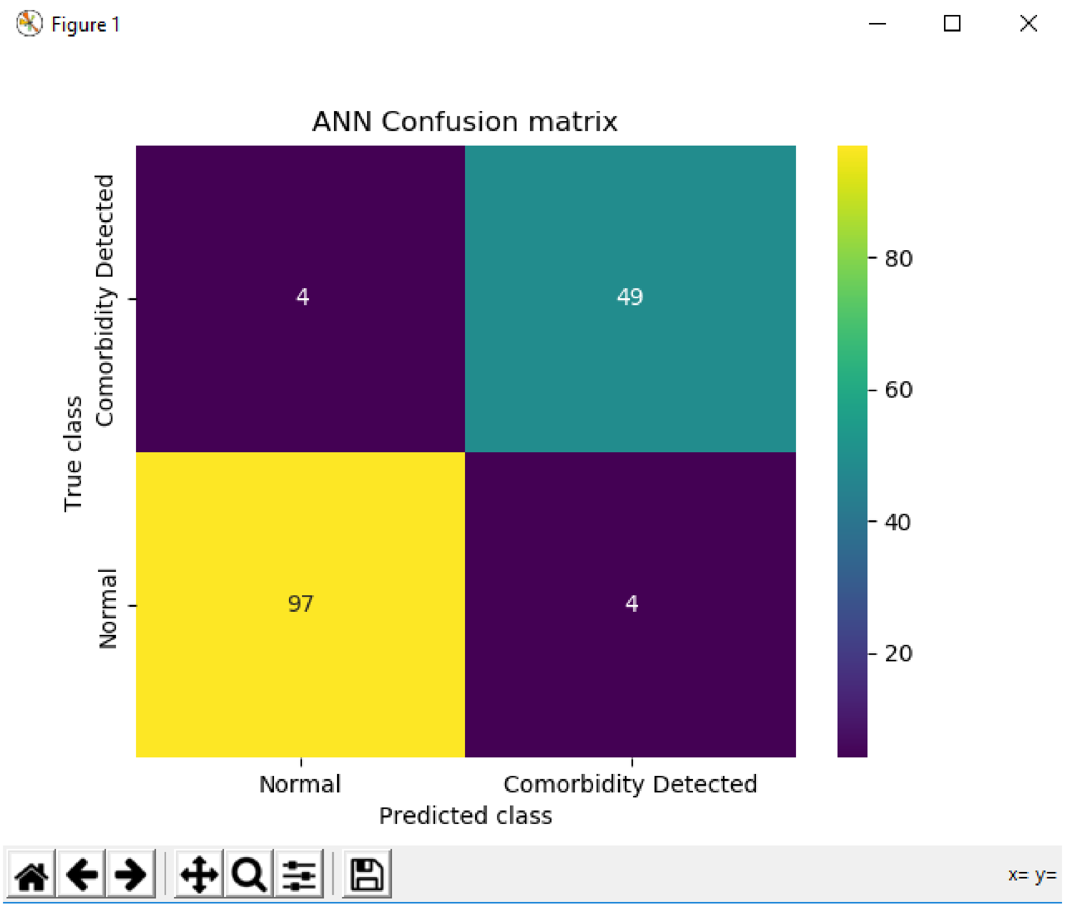


Figure 7: Displays the Confusion matrix of ANN model.

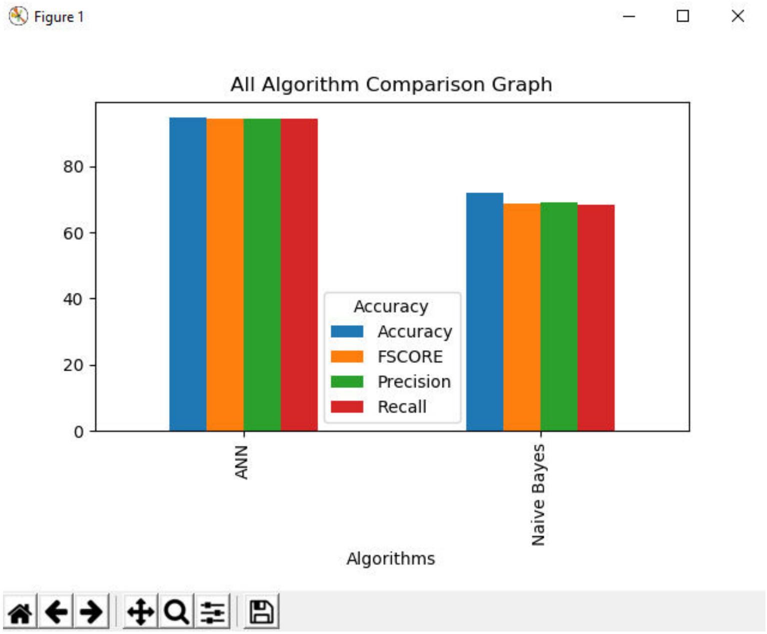
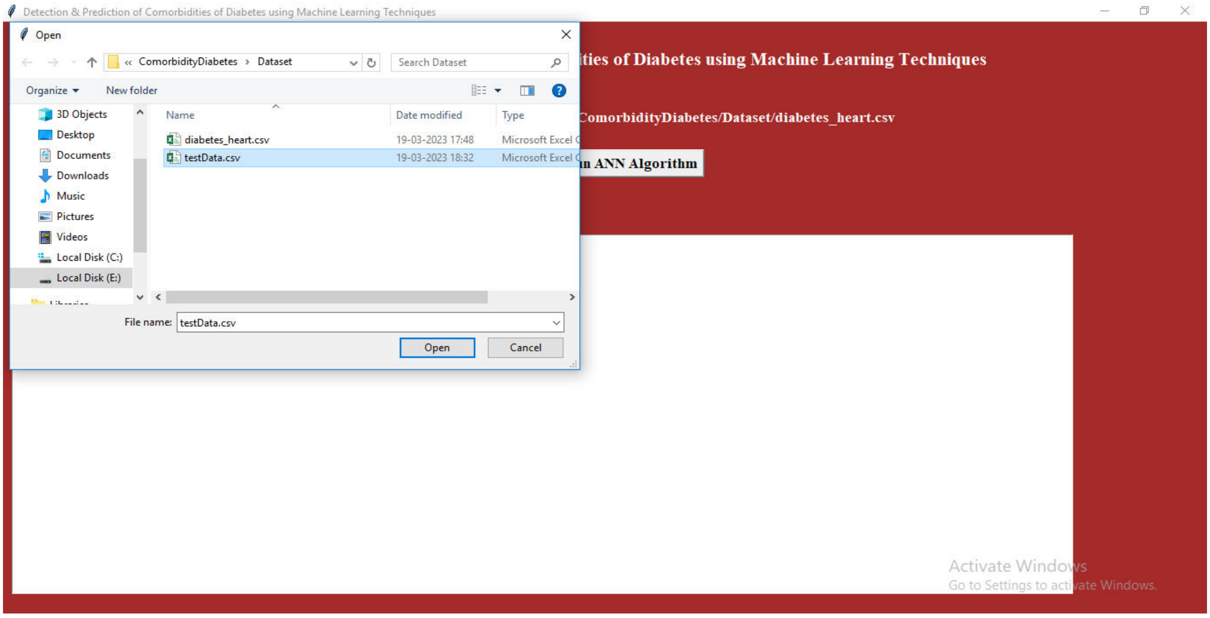
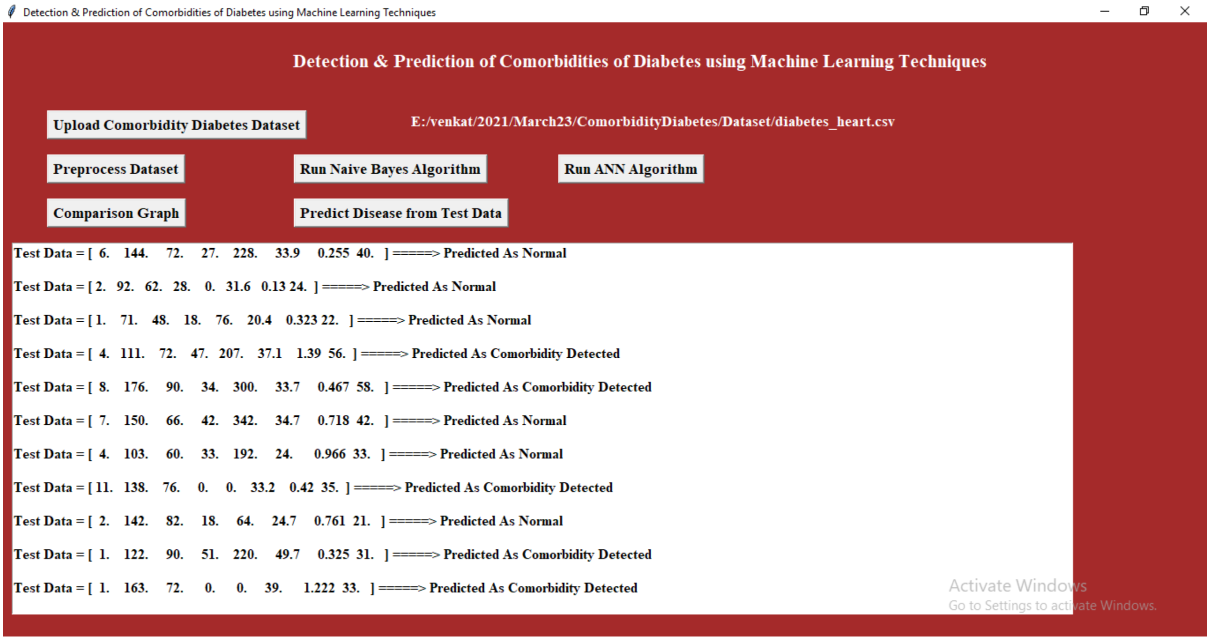


Figure 8: Represents the performance comparison graph of each model.


Detection & Prediction of Comorbidities of Diabetes using Machine Learning Techniques
Open
« ComorbidityDiabetes › Dataset
Search Dataset
Organize
New folder
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Name
Date modified
Type
Open
Cancel
ities of Diabetes using Machine Learning Techniques
n ANN Algorithm
Activate Windows
Go to Settings to activate Windows.

```
diabetes_heart.csv    19-03-2023 17:48    Microsoft Excel C
testData.csv          19-03-2023 18:32    Microsoft Excel C
File name: testData.csv
ComorbidityDiabetes/Dataset/diabetes_heart.csv
```

Figure 9: Displays the uploading of test data for model prediction.


Detection & Prediction of Comorbidities of Diabetes using Machine Learning Techniques
Detection & Prediction of Comorbidities of Diabetes using Machine Learning Techniques
Upload Comorbidity Diabetes Dataset
Preprocess Dataset
Run Naive Bayes Algorithm
Run ANN Algorithm
Comparison Graph
Predict Disease from Test Data
Activate Windows
Go to Settings to activate Windows.

```
E:/venkat/2021/March23/ComorbidityDiabetes/Dataset/diabetes_heart.csv

Test Data = [  6.   144.    72.    27.   228.    33.9    0.255  40.   ] =====> Predicted As Normal
Test Data = [ 2.   92.   62.   28.    0.   31.6   0.13 24.  ] =====> Predicted As Normal
Test Data = [ 1.    71.    48.    18.    76.    20.4    0.323 22.   ] =====> Predicted As Normal
Test Data = [  4.   111.    72.    47.   207.    37.1    1.39  56.  ] =====> Predicted As Comorbidity Detected
Test Data = [  8.   176.    90.    34.   300.    33.7    0.467  58.   ] =====> Predicted As Comorbidity Detected
Test Data = [  7.   150.    66.    42.   342.    34.7    0.718  42.   ] =====> Predicted As Normal
Test Data = [  4.   103.    60.    33.   192.    24.     0.966  33.   ] =====> Predicted As Normal
Test Data = [ 11.   138.    76.    0.    0.    33.2    0.42  35.  ] =====> Predicted As Comorbidity Detected
Test Data = [  2.   142.    82.    18.    64.    24.7    0.761  21.   ] =====> Predicted As Normal
Test Data = [  1.   122.    90.    51.   220.    49.7    0.325  31.   ] =====> Predicted As Comorbidity Detected
Test Data = [  1.   163.    72.     0.     0.    39.     1.222  33.   ] =====> Predicted As Comorbidity Detected
```

Figure 10: Displays the model prediction on test data.

5. CONCLUSION

The integration of machine learning into healthcare marks a significant advancement in the identification and prediction of comorbidities among individuals with diabetes. Unlike traditional approaches that rely heavily on clinical judgment and manual analysis, machine learning models adopt a data-driven methodology capable of uncovering intricate patterns and relationships within large-scale datasets comprising patient demographics, medical histories, and diagnostic results. The urgent need for accurate and timely comorbidity diagnosis underscores the value of these models in enhancing treatment strategies and improving patient outcomes. By efficiently processing vast and complex information, machine learning enables earlier detection and more effective management of comorbid conditions, paving the way for personalized healthcare interventions that can substantially improve the quality of life for diabetic patients.

REFERENCE

1) Deo, R.C. Machine Learning in Medicine. Circulation 2015, 132, 1920–1930.
2) Yuvaraj, N.; SriPreethaa, K.R. Diabetes prediction in healthcare systems using machine learning algorithms on Hadoop cluster. Clust. Comput. 2017, 22, 1–9.
3) Tafa, Z.; Pervetica, N.; Karahoda, B. An intelligent system for diabetes prediction. In Proceedings of the 2015 4th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 14–18 June 2015; pp. 378–382.
4) Sisodia, D.; Sisodia, D.S. Prediction of Diabetes using Classification Algorithms. Procedia Comput. Sci. 2018, 132, 1578–1585.
5) Mercaldo, F.; Nardone, V.; Santone, A. Diabetes Mellitus Affected Patients Classification and Diagnosis through Machine Learning Techniques. Procedia Comput. Sci. 2017, 112, 2519–2528.
6) Negi, A.; Jaiswal, V. A first attempt to develop a diabetes prediction method based on different global datasets. In Proceedings of the 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, India, 22–24 December 2016; pp. 237–241.
7) Olaniyi, E.O.; Adnan, K. Onset diabetes diagnosis using artificial neural network. Int. J. Sci. Eng. Res. 2014, 5, 754–759.
8) Soltani, Z.; Jafarian, A. A New Artificial Neural Networks Approach for Diagnosing Diabetes Disease Type II. Int. J. Adv. Comput. Sci. Appl. 2016, 7, 89–94.
9) Somnath, R.; Suvojit, M.; Sanket, B.; Riyanka, K.; Priti, G.; Sayantan, M.; Subhas, B. Prediction of Diabetes Type-II Using a Two-Class Neural Network. In Proceedings of the 2017 International Conference on Computational Intelligence, Communications, and Business Analytics, Kolkata, India, 24–25 March 2017; pp. 65–71.
10) Mamuda, M.; Sathasivam, S. Predicting the survival of diabetes using neural network. In Proceedings of the AIP Conference Proceedings, Bydgoszcz, Poland, 9–11 May 2017; Volume 1870, pp. 40–46.
11) Mohebbi, A.; Aradóttir, T.B.; Johansen, A.R.; Bengtsson, H.; Fraccaro, M.; Mørup, M. A deep learning approach to adherence detection for type 2 diabetics. In Proceedings of the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jeju, Korea, 11–15 July 2017; pp. 2896–2899.
12) Miotto, R.; Li, L.; Kidd, B.A.; Dudley, J.T. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. Sci. Rep. 2016, 6, 26094.
13) Pham, T.; Tran, T.; Phung, D.; Venkatesh, S. Predicting healthcare trajectories from medical records: A deep learning approach. J. Biomed. Inform. 2017, 69, 218–229.
14) Balaji, H.; Iyengar, N.; Caytiles, R.D. Optimal Predictive analytics of Pima Diabetics using Deep Learning. Int. J. Database Theory Appl. 2017, 10, 47–62.
15) Lekha, S.; Suchetha, M. Real-Time Non-Invasive Detection and Classification of Diabetes Using Modified Convolution Neural Network. IEEE J. Biomed. Health Inform. 2018, 22, 1630–1636.

16) Askarzadeh, A.; Rezazadeh, A. Artificial neural network training using a new efficient optimization algorithm. Appl. Soft Comput. 2013, 13, 1206–1213.