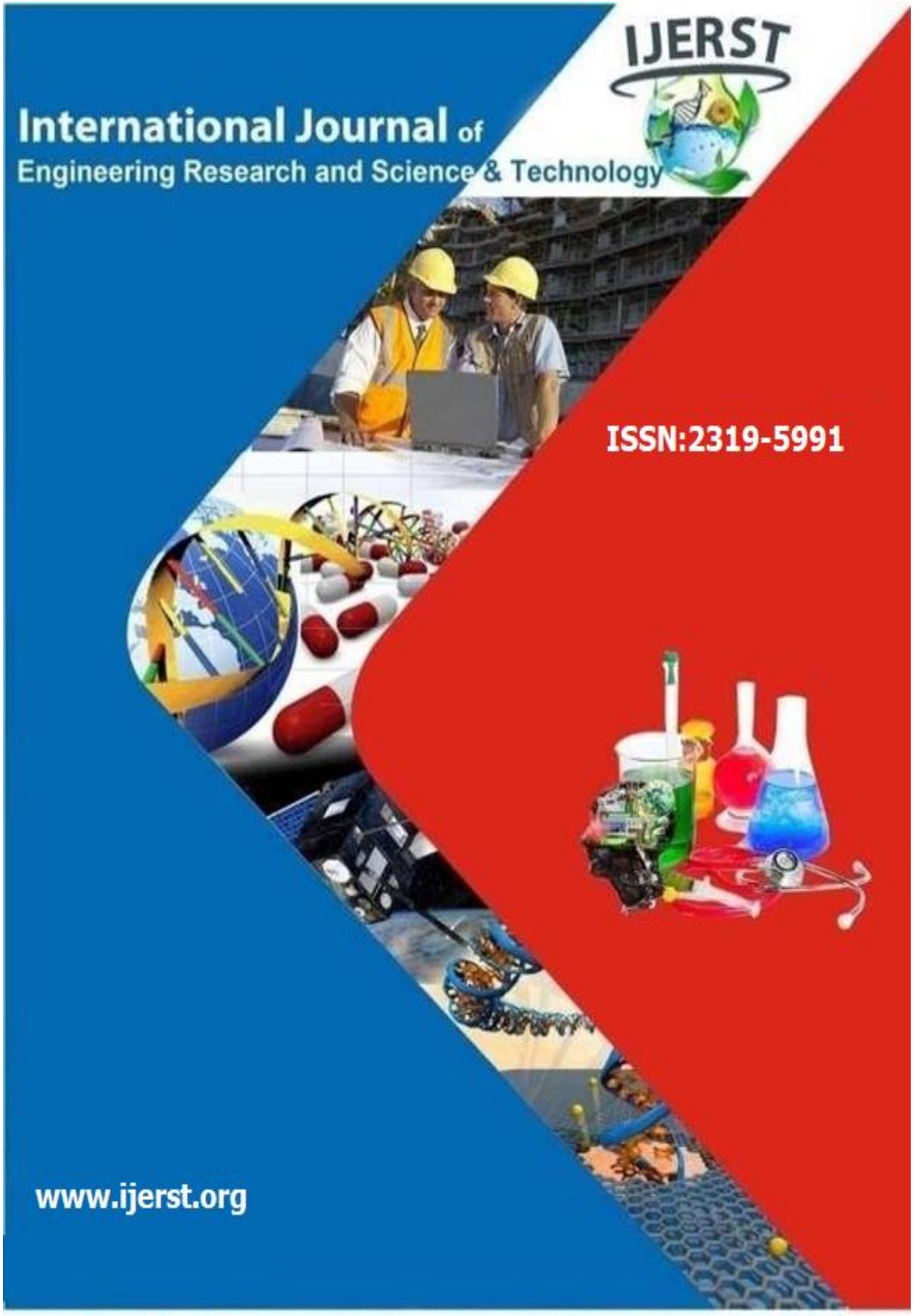


E-mail: editor@ijerst.org or ijerst.editor@gmail.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

DETECTION & PREDICTION OF COMORBIDITIES OF DIABETES USING MACHINE LEARNING

MS.M. ANITHA[1], K.BABY RAMYA[2], E. SRAVANI[3]

[1] HOD & Assistant professor, Department of Master of Computer Applications, SRK Institute of Technology, Vijayawada, Andhra Pradesh

[2]Assistant professor, Department of Master of Computer Applications, SRK Institute of Technology, Vijayawada, Andhra Pradesh

[3] MCA Student, Department of Master of Computer Applications, SRK Institute of Technology, Vijayawada, Andhra Pradesh

ABSTRACT

High blood sugar levels that persist for a long time are the hallmark of diabetes, a metabolic illness. A number of health problems, or co morbidities, might develop as a result. These can include things like neuropathy, cardiovascular disease, renal disease, and kidney failure. In order to intervene quickly and enhance patient outcomes, it is essential to detect and anticipate co morbidities in people with diabetes. Clinical observations and statistical evaluations of patient data were the backbone of early attempts to manage diabetic co morbidities. The intricacy and diversity of co morbidity patterns were beyond the capabilities of these techniques, notwithstanding their value. A sea change occurred in this field with the introduction of machine learning, which allowed for the creation of more complex and precise prediction models. One challenge in using machine learning to the detection and prediction of diabetes co morbidities is the need to use data from diabetes patients to construct models that can determine the probability of acquiring certain co morbidities. This work necessitates the analysis of numerous datasets in order to provide healthcare professionals with meaningful insights. These datasets may include patient demographics, medical history, test findings, and even genetic information. Healthcare providers used to rely heavily on manual analysis for diabetes co morbidity diagnosis and prediction. Human subjectivity and the inefficiency of processing enormous quantities of data severely constrained this technique, even if they depended on their knowledge and experience. Machine learning models excel in detecting complex patterns, which it failed to do as well. Co morbidity diagnosis and prediction in diabetic patients must be done accurately and promptly. People with diabetes have a much worse quality of life due to co morbidities, and better treatment options are available with earlier intervention. The promise of machine learning lies in its ability to sift through massive datasets in search of hidden links and patterns that humans may miss. More tailored and efficient healthcare interventions may result from this. Massive volumes of patient data may be analysed with the help of ML algorithms to find patterns and predictive variables linked to certain co morbidities. The predicted accuracy of these models may be enhanced over time as they learn from past data, incorporate new information, and adapt. In this setting, machine learning's use is a huge step forward in personalised medicine, with the ability to alleviate co morbidities and make people's lives with diabetes much better.

KEYWORDS: Machine Learning, Diabetes, Health care.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

1. INTRODUCTION

1.1 History

From first clinical observations and statistical analysis to the revolutionary incorporation of machine learning methods, the historical trajectory of managing diabetic comorbidities demonstrates a development. In the beginning, healthcare providers had to rely on their knowledge and experience to manually analyse patient data in order to identify and anticipate comorbidities. But there were problems with this method from the start, such how subjective it was, how slow it was to handle large datasets, and how it couldn't pick up on complex patterns. There was a sea change when machine learning came along; now, we can build prediction models that are both complex and accurate. This was a huge change from the old ways since machine learning could use all sorts of data, including genetic information, medical records, test findings, and patient demographics. The challenge then shifted to finding ways to use these datasets to train algorithms that might predict which diabetic patients were more likely to have certain co-occurring disorders. Because comorbidities have such a significant influence on diabetes patients' quality of life, early intervention is crucial for developing successful treatment methods, making accurate and rapid identification of the disease a top priority. Machine learning changed the game by analysing massive datasets in ways that humans can't, revealing hidden patterns and links. This breakthrough is a huge step forward for personalised medicine and might improve diabetic patients' quality of life by making healthcare more proactive and accurate, therefore lowering the prevalence of comorbidities.

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. Co morbidity 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, co morbidity 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 co morbidity 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

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

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 co morbidity 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.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

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

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

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

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

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

3.1 Overview

The Python source code describes machine learning research that uses many methods to identify and predict diabetic comorbidities.

The procedures are outlined here in great detail:

- At the start of the project, the user is given the option to submit the dataset including information on co morbidity diabetes. A file dialogue is included in the Tkinter GUI, which allows the user to choose the dataset file. The file location is shown on the GUI after loading, and the first few rows of the dataset are shown.
- This stage is called dataset preparation and it is initiated by the user once the dataset has been loaded. To do this, we must replace all null values with zeros. In order to facilitate further processing, the dataset is transformed into NumPy arrays. The dataset is normalised using conventional scaling, and features and labels are segregated.
- Using the train_test_split function from Scikit-Learn, the normalised dataset is divided into training and testing sets. Dataset statistics, including total records and training/testing splits, are included in the script.
- The Naive Bayes method is used for illness prediction in the project. Predictions are produced using the test set after training the Gaussian Naive Bayes model on the training set. A number of performance indicators are computed and shown, including F1-score, recall, accuracy, and precision. For the purpose of visualising the results, a confusion matrix is also produced.
- A model that uses an Artificial Neural Network (ANN) to forecast diseases is put into place. The script constructs a multi-layer neural network using Keras. Using the training set, the model is trained and the weights are stored for later use. Instead, the model weights are loaded if they exist. The test set is used to assess the ANN model, and performance metrics are presented in a manner similar to those of the Naive Bayes model.
- Various performance measures, such as accuracy, precision, recall, and F1-score, are computed and shown by the script for both Naive Bayes and ANN models in the comparison graph. To

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

illustrate the relative merits of the two methods, a comparison graph is created.

- The project has now made it possible for the user to forecast illnesses using test data. The ANN model makes illness outcome predictions for each entry once the user uploads fresh test data. The graphical user interface (GUI) shows the outcomes, which include the test data and the projected classes.

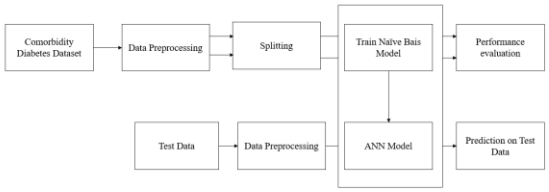


Figure 3.1: Block Diagram of Proposed Model.

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.

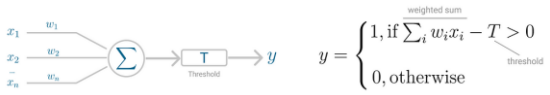


Fig. 3.2: Perceptron neuron model (left) and threshold logic (right).

Threshold 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

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

discovers the separating hyperplane. As shown below, the perceptron loss function:

$$\underbrace{D(w,c)}_{\text{distance}} = -\sum_{i \in \mathrm{M}} \overbrace{y_i}^{\text{output}} (x_i w_i + c)$$

misclassified observations

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.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

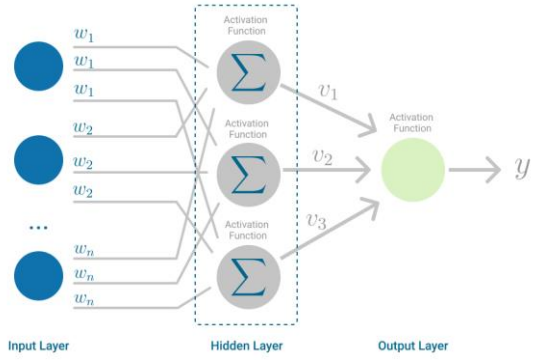


Fig. 3.4: Architecture of ANN.

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:

$$\Delta_w(t) = -\varepsilon \frac{dE}{dw_{(t)}} + \alpha \Delta_{w(t-1)}$$

Bias, Error, Learning Rate, Gradient Current Iteration, Weight vector, 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.

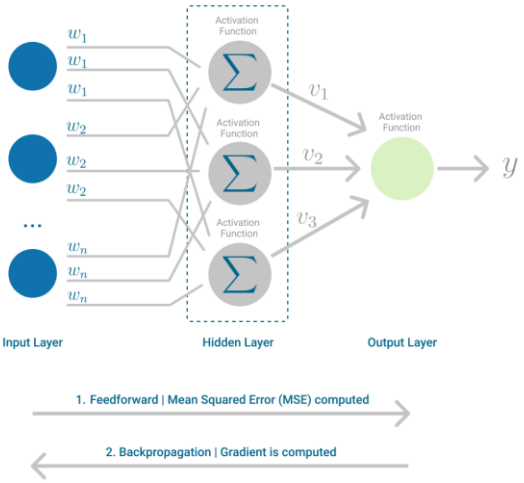


Fig. 3.5: ANN, highlighting the Feedforward and Backpropagation steps.

4. RESULTS

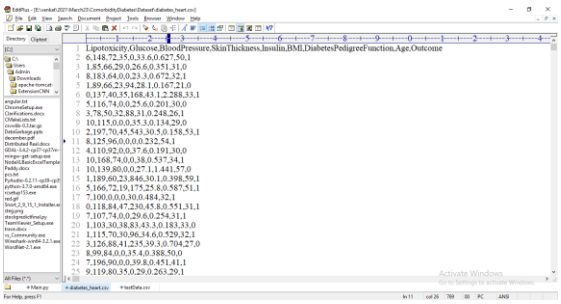
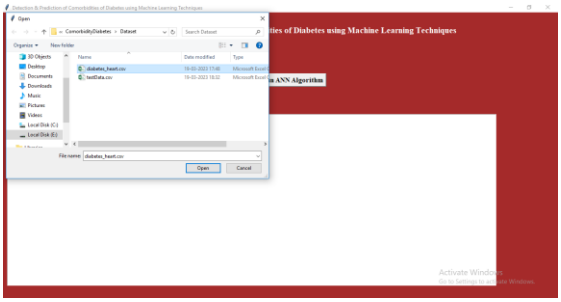


Figure 1: Displays the dataset.

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



International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

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

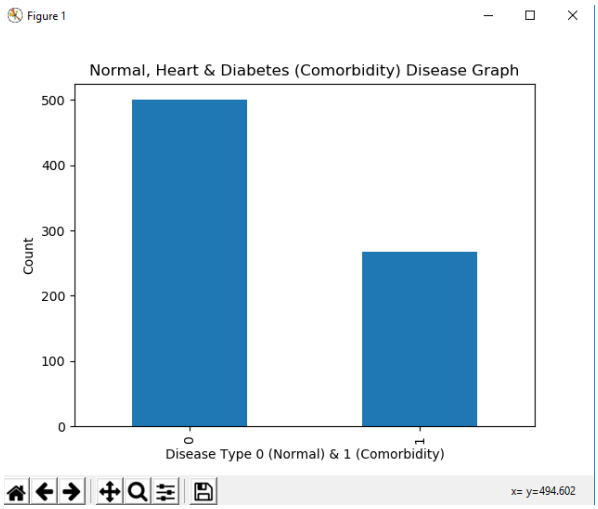


Figure 4: Shows the count plot of each class in the dataset.

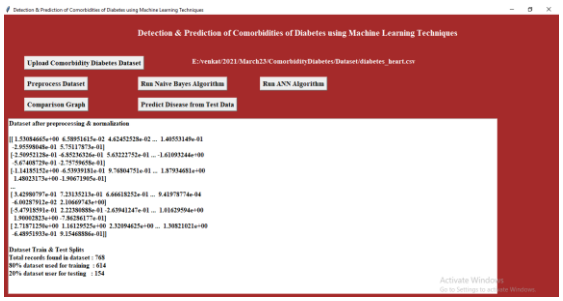


Figure 5: Displays the dataset preprocessing and normalization.

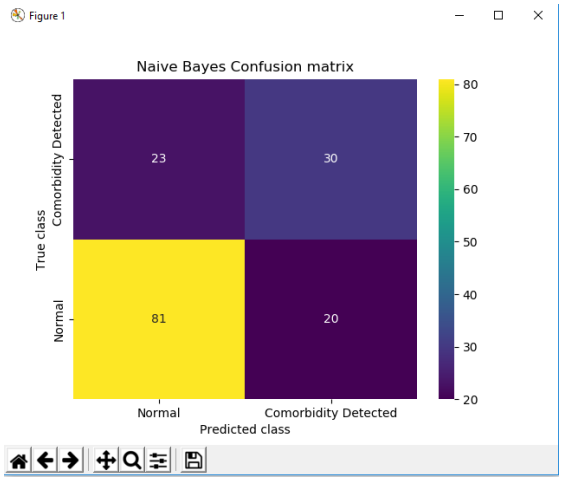


Figure 6: Presents the confusion matrix of Naïve Bayes Model.

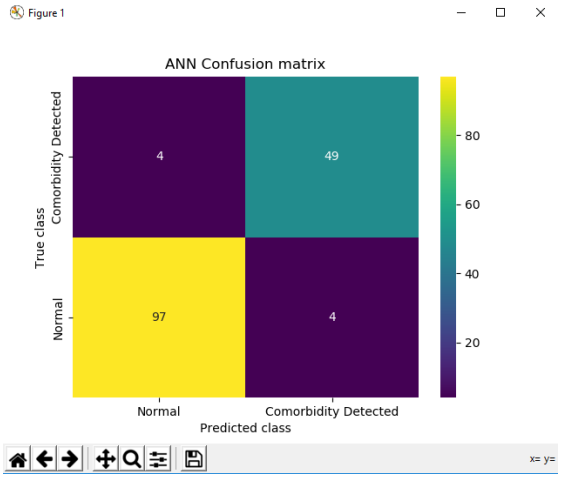


Figure 7: Displays the Confusion matrix of ANN model.

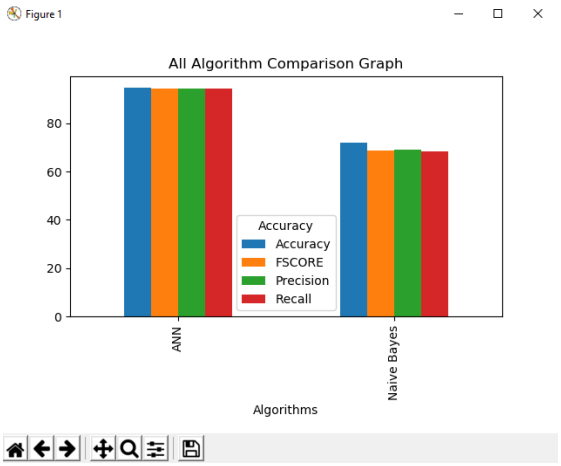


Figure 8: Represents the performance comparison graph of each model.

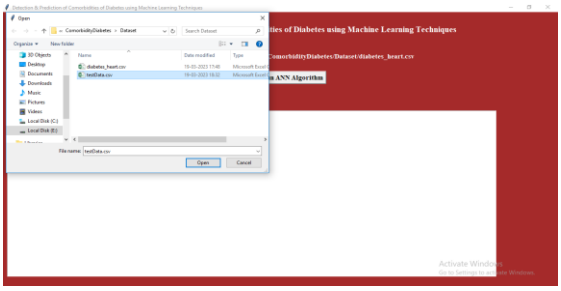


Figure 9: Displays the uploading of test data for model prediction.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

Figure 10: Displays the model prediction on test data.

5. CONCLUSION

Finally, a major step forward in healthcare has been the use of machine learning for the purpose of co morbidity identification and prediction in diabetic persons. Machine learning models provide a more complex and data-driven approach than the old ways, which depended on human analysis and clinical observations. These programs may find complex links and patterns that humans would miss by analysing large datasets that include patients' demographics, medical records, and test results. Improving treatment options and patient outcomes highlights the critical need for precise and rapid co morbidity diagnosis in diabetes. Early and effective treatment of comorbid illnesses may be achieved via the use of machine learning methods, which analyse massive and complicated information to give a road to more personalised healthcare interventions.

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

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

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.