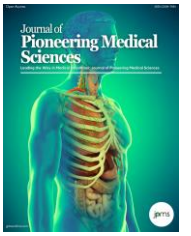
Journal of Pioneering Medical Sciences
Received February 28, 2024 Accept March 22, 2024 Publish July 30, 2024.
Volume 13, Issue 2, Pages 75-87

DOI https://doi.org/10.61091/jpms202413413



Developing a Neural Network Model for Type 2 Diabetes Detection

Noha Alsulami[1,*], Shahenda Sarhan[2], Miada Almasre[1] and Wafaa Alsaggaf[1]
[1]Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University Jeddah, Saudi Arabia.
[2]Computer Science Department, Faculty of Computers and Information Sciences, Mansoura University, Mansoura, Egypt.

Corresponding author: Noha Alsulami (e-mail: nmuttlaqalsulami@stu.kau.edu.sa).

Abstract Worldwide, the healthcare system is greatly impacted by the changing requirements of the people. Diabetes is a long-lasting condition that can lead to serious complications if not controlled correctly. It is divided into Type 1 (TID) and Type 2 (T2D) diabetes. Research shows that almost 90% of Diabetes cases are T2D, with TID making up around 10% of all Diabetes cases. This paper suggests a Rough-Neuro classification model for identifying Type 2 Diabetes, which includes a two-stage process. The approach includes utilizing rough sets JohnsonReducer to eliminate unnecessary features or characteristics and multilayer perceptron for illness categorization. The suggested technique seeks to reduce the amount of input characteristics, which results in a reduction in the time needed to train the neural network and the storage space required. The findings show that decreasing the amount of input characteristics results in a lower neural network training time, enhances model performance, and reduces storage needs by 63%. It is worth mentioning that a smaller neural network with only seven hidden layers, trained for 1000 epochs with a learning rate of 0.01, attained the best performance, but time and storage were much decreased.

Key Words Diabetes, Type 2 diabetes, Rough set, Reducts, Attribute reduction, Neural networks

1. Introduction

Diabetes is a recognized chronic condition, also referred to as diabetes mellitus, that is associated with various negative health outcomes like stroke, chronic kidney failure, heart attack, and diabetic foot syndrome, among others [1], [2]. As per forecasts from the WHO (World Health Organization) [3], diabetes is expected to become the seventh leading cause of death by 2030. Additionally, as per the International Diabetes Federation, there is a projected increase in the number of individuals with diabetes over the next 26 years, reaching 693 million in comparison to 451 million worldwide in 2017 [4]. Diabetes is described as a persistent metabolic disorder that gives rise to variations in blood glucose levels, typically classified into two primary forms: Type 1 and Type 2. TID is a result of inadequate insulin production in the body, while T2D occurs due to the body's inability to utilize the insulin it produces. Type 1 diabetes makes up around 10% of diabetes cases, with Type 2 diabetes accounting for the remaining 90% [3]. T2D is increasingly becoming a widespread issue for the medical field [1]. Even though the exact reason for diabetes remains a mystery, experts suggest that it may result from a combination of genetic and environmental influences. Diabetes poses a significant threat due to its inability to be cured. It is believed that medications and specific drugs can manage the condition. In addition, early detection of diabetes is crucial for minimizing complications and serious health problems [5].

Rough Set theory is a modern technique for managing uncertainty. It helps identify data dependencies, categorize and rank objects, assess the significance of features, reduce redundancies, and classify data types. Furthermore, it is utilized for retrieving rules from databases, with one benefit being the generation of comprehensible if-then rules. These bases have the potential to discover types of information that were previously unknown. Additionally, it serves as a classifier for specimens that are not observable. Relying solely on the data provided in personal information, rough set analysis does not require external factors. Furthermore, another significant benefit is that rough set theory can determine the completeness of data through a straightforward evaluation. Additionally, it offers guidance on the necessary items in case the information is not complete [6].

In addition, even if the data is incomplete, rough sets can still detect data duplicates and determine the minimum

information needed for evaluation [6]. This important property is crucial when there is limited domain knowledge for applications or when collecting data is expensive or time-consuming. It ensures that the data collected is sufficient to create a reliable classification model, saving time and effort while maintaining accuracy in gathering more information about the objects [7].

The Rough set theory is applied across various fields to address challenges related to classification, feature selection, decision-making, and knowledge discovery. Some domains that apply rough set theory include: Medicine (the rough set theory is applied in the analysis of medical data for tasks such as diagnosing diseases, distinguishing patterns, and making decisions based on medical data) [7], [8]. Researchers have employed rough set theory to examine financial data for objectives such as evaluating credit risks and forecasting financial results [9]. In addition, rough set theory is utilized to execute image processing tasks, including image segmentation and feature selection [10]. Researchers utilize the rough set theory for pattern recognition, object classification, and feature selection [11]. It has been applied in various fields such as real-time strategy games for categorizing opponent behavior [6], predicting the decrease in forest fire risks [12], and aiding in decision-making for security forces' operations [13].

Reducing data can help condense a large dataset into a smaller size without compromising the original data's integrity [14]. Reducing features keeps the original characteristics intact while selecting a subset that accurately forecasts the intended class variable [15]. In addition, neural network classifiers face various challenges, including training expenses, increased storage, and time consumption in neural networks with larger input dimensions. This paper aims to utilize rough set theory due to its ability to reduce attributes. Furthermore, it provides a straightforward, concise, and easily comprehensible explanation. This suggests that there are no difficulties, intricacies, or undisclosed stages to consider. It pertains to an economical or budget-conscious approach to solving a problem or meeting a need with solutions. Consequently, it reduces the amount of input features and shortens the time needed for network training. This paper presents a model that merges an approximate set reduction algorithm with a neural network classifier. Here is the paper's structure: Section 2 presents established models for identifying type 2 diabetes. Section 3 provides an overview of the solution that has been put forward. Section 4 exhibits the findings of the present investigation. Section 5 brings the paper to a close.

2. Literature Review

Reducing features, as discussed earlier, helps decrease the number of input features, resulting in reduced training time and storage needs, while also enhancing network performance. The goal of feature selection is to enhance prediction accuracy and gain deeper insights into the data analysis process. This document offers a summary of various classification techniques. Many of these classification techniques rely on standard algorithms to assess the effectiveness of the features in the dataset. Choosing the best features will decrease the time complexity and space while also enhancing the accuracy of classification. This section discusses the current research in literature on detecting diabetes, focusing on type 2 diabetes, to examine the impact of utilizing feature reduction or selection.

In a study by Kakoly et. al. [16], a questionnaire was created and distributed among both urban and rural communities in Bangladesh. Data from 738 subjects was gathered and prepared by addressing missing values and outliers. Next, two techniques were employed to select features: Information Gain (IG) and Principal Component Analysis (PCA). Subsequently, these features were input into five distinct classifiers: Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), k-nearest neighbors (KNN) and Logistic Regression (LR). The outcomes demonstrate an accuracy rate exceeding 82.2%, accompanied by an Area Under the Curve (AUC) value of 87.2%.

Li et. al.'s [17] study used three feature selection techniques to improve the Pima Indian Diabetes Dataset (PIDD) classification performance. In conjunction with the K-means clustering technique, the researchers investigated several permutations of the genetic algorithm (GA), particle swarm optimization (PSO), and harmony search (HR). By means of the GA-Kmeans amalgamation, it was ascertained that Blood pressure, Insulin, and Age assumed a pivotal role in the classification. Likewise, the combination of GA-PSO and K-means showed that Glucose, Blood pressure, Insulin, and BMI were significant factors. Ultimately, the HR-Kmeans combination identified blood pressure, insulin, and glucose as critical variables. The KNN classifier was used by the researchers to categorize diabetes cases. Surprisingly, the suggested feature selection method combinations performed better than earlier results on the same dataset. With an accuracy of 91.65%, the HR-Kmean hybrid combination outperformed the others, demonstrating how much it improved classification performance.

In addition, Saxena et. al. [18] applied three different methods to select features in the PIDD dataset to identify T2D: IG, correlation attribute evaluation, and PCA. The dataset has been pre-processed by eliminating the outliers and replacing the missing values with the mean value. Afterward, they utilized three different methods to select either 4 or 6 features. This study employed four machine learning algorithms: DT, KNN, multilayer perceptron (MLP), and RF. According to the findings, random forest achieved the highest accuracy of 79.8%.

Rahman et. al. [2] developed a Convolutional Long Short-term Memory (Conv-LSTM) model to predict diabetes using the PIDD dataset. For comparison, the CNN (Convolutional Neural Network), CNN-LSTM, and Traditional LSTM (T-LSTM) models were employed. The Boruta algorithm chose the following features: age, BMI, glucose, blood pressure, and insulin. The median was utilized by the authors to fill in the missing values when they employed grid search for hyperparameter optimization. Embedding, Conv-LSTM, and dense layers made up the model. After splitting the dataset into two categories, the suggested model's accuracy was 91.38%, and after five-fold cross-validation, it was 97.26%. However, performance suffered due to the model's intricacy. Kumar et.

al. used a Deep Neural Network (DNN) classifier to predict T2D through an unsupervised learning method [19]. The model's performance was assessed using PIDD, and the dataset was pre-processed by eliminating unclear or missing data. In order to improve the model, certain features were chosen according to their importance score, such as BMI, Glucose, Age, and Diabetes pedigree function. These characteristics were subsequently utilized to train the DNN. The model has four nodes in the input layer, one node in the output layer, and three hidden layers with 20, 10, and 10 nodes each. The results showed that the model outperformed previous studies in the field, achieving a precision of 98.16%. Nevertheless, this model is constrained by its substantial computational expenses resulting from DNN processing.

In addition, Zhou et. al. [20] utilized a Deep Learning model for Predicting Diabetes called DLPD to make predictions about Diabetes. This model has the ability to forecast potential forms of diabetes that may develop later on. The model was developed using DNN and assessed with reference to the Diabetes Type Dataset (DTD) and the PIDD. The plan is divided into four phases. First prepare the dataset, then build and train the DLPD model, run the normal output and tune the hyperparameters. The authors initially divided the dataset into training (70%), validation (15%), and testing (15%) data for prioritization. In the second stage of the proposed model, there are three layers. The input layer simply passes the dataset's features to the hidden layers without performing any computations on them. There is no limit to the number of hidden layers. The dataset is processed within the hidden layers, and the outcomes are then transferred to the output layer. During the third phase, dropouts are controlled to prevent overfitting. In order to create an accurate prediction model for DNN, the authors adjusted certain parameters prior to applying the binary cross-entropy loss function. The results from the experiment showed that the proposed model performed better. Nevertheless, there is no comparison to the current related works.

Naz and Ahuja [21], utilized three supervised learning algorithms: DT, Artificial Neural Network (ANN), and Naive Bayes (NB). In order to identify diabetes, the researchers utilized deep learning, specifically a layered feed-forward neural network. They utilized stochastic gradient descent with back-propagation for the training process. PIDD was used to measure performance in this study. To ensure the validity of the research, the data was divided into testing and training. The model proposed in this study consists of an input layer for input data, two hidden layers for processing the data set, and an output layer for prediction. Experimental results show that the maximum accuracy of the multilayer feedforward perceptron model reaches 98.07%. However, the dataset was not pre-processed by the authors.

In addition, Lukmanto et. al. in [22] introduced a framework for identifying T2D. This model was evaluated using PIDD. The data was first processed by removing features that contained many missing data, such as skin thickness and insulin. The F-Score selection process is then used to select specific features from the PIDD database. Only blood sugar and body mass index were used in the classification process. The data is divided into two as training and testing, 87% is used for training and 13% for testing. Data classification using fuzzy support vector machine model. According to the results, the proposed model achieved an accuracy of 89.02%.

Prabhu and Selvabharathi [23] developed a Deep Belief Neural (DBN) network model for diabetes detection. The research utilized PIDD to assess the performance of the model. The model consists of three main stages: pre-processing, pre-training using DBN, and fine-tuning. Normalization is a method utilized to prepare the dataset before processing. The appropriate values are selected from the training database using PCA. Normalization is typically carried out during the pre-processing phase in machine learning, particularly prior to utilizing PCA. Normalization is not a built-in aspect of PCA, yet it is crucial to pre-process data prior to employing PCA or similar methods. In the pretraining phase, the DBN consists of an output layer, an input layer, and three hidden layers, each utilizing a Rectified Linear Unit (ReLU) as the activation function. The classifier is developed further during the fine-tuning phase based on the results obtained from the initial training phase. Based on the experimental findings, the new model performs better than traditional models like NB, DT, LR, SVM, and RF. They failed to utilize an optimization method to address overfitting.

Kumar and Manjula utilized the Keras toolkit to develop an MLP network for diabetes detection [24]. The evaluation of performance in this study was conducted using PIDD. The author transformed categorical data and independent variables to organize the data. There are two layers in the model: IL (input layer) and OL (output layer). IL uses the ReLU activation function, while OL uses the sigmoid activation function. According to the research results, the accuracy of the sample request is 86.67%. They chose not to use dropout as a way to avoid overfitting.

A deep wide and deep learning model, a deep feedforward neural network, and the power of an established linear model were combined in [25], Nguyen et. al.'s model, to improve the model's overall execution by eliminating features related to glucose or insulin. To detect T2D, the proposed methodology relies on electronic health information for the United States population. Three groups were created from the 1312 features. The categories include fixed and basic features like blood pressure, sex, BMI, and age; crossed features including the selection of top diagnosis and medication characteristics; and adjustable features such diagnostic features dependent on laboratory tests and medication. The proposed model employed the Synthetic Minority Oversampling Technique (SMOTE) at 150 and 300 percent for each fold of the cross-validated training to evaluate the experiment's outcomes. Three sets of features are represented by the embeddings of the hidden layers in the deep component. 151 input features are available for diagnosis, 134 input features are available for treatment, and 80 input features are available for laboratory testing. In order to improve the learning process from a sparse binary vector to a dense 16-dimensional vector, each embedding was done using independent shallow layers. There were 256 and 128 neurons in the hidden layers, respectively. To construct a 1439-dimensional vector, these were added to the broader section in the last layer that contained the intersecting features together with the result of the deep component. A single layer with a 128-to-1 layer and a logistic activation function was used in the finalization of the framework. ReLU served as the activation

function in the other layers. The average of the output probabilities from the top 10 models was computed to construct the most recent predictive model for the start of T2D, which was then compared to a threshold of 0.5 to indicate diabetes. The results indicated that the performance of the proposed model using the identical dataset outperformed other machine learning algorithms. The study faced challenges due to the dataset's high dimensions and sparsity. Additionally, the wide and deep model could not predict certain important risk factors within the model.

In addition, Kannadasan et. al. in [26] created a deep neural network model to forecast T2D by utilizing the stacked autoencoder. The performance of the suggested model was assessed using various metrics such as recall, F1-score, precision, accuracy, and specificity with the help of PIDD. Various features are extracted from the dataset through a stacked autoencoder, followed by categorizing the dataset using a softmax layer. Tests were conducted in two separate scenarios to assess efficiency. The initial situation required fine-tuning, while the subsequent one did not. During the last stage of classification, fine-tuning is employed to enhance performance by utilizing backpropagation with the training dataset under supervision. The suggested model was compared with various other models and cutting-edge techniques in the field. The results showed that the proposed model surpasses previous models, achieving an accuracy of 86.26%. The dataset was not pre-processed by the authors.

Deshmukh and Fadewar [27] utilized a hybrid fuzzy deep learning method that relied on deep CNN to identify diabetes. This model depends on converting data into a matrix to meet the requirements of CNN. Following the process of fuzzification, every data point in this framework gets converted into a 5 × 5 matrix. The value represented is the rows of the matrix and the characteristics as the columns of the matrix. This study uses PIDD to evaluate performance. Regarding CNN, the network was made up of complex spiral layers with a 3 × 3 kernel size and pooling layers with a 2 × 2 kernel size. The results show that utilizing CNN with fuzzification is more effective than traditional neural networks for identifying diabetes. However, the dataset was not pre-processed by the authors.

Ashiquzzaman et. al. [28] created a deep neural network model for detecting diabetes along with their other research projects. In this research, PIDD was utilized. They addressed the issue of overfitting in this model by incorporating a regularization layer called Dropout. The model proposed in this paper includes an output layer, an input layer, two fully connected layers (FCL), and two dropout layers. The process starts by inputting data into the input layer. After that, every FCL contains a dropout layer. The initial FCL consists of 64 neurons and uses ELU as an activation function, while the second FCL consists of 23 neurons and also uses ELU as an activation function. The final layer consists of a single neuron that uses Softplus as an activation function to produce a decision. The Backpropagation algorithm is used to improve the model. Based on the findings, the model surpasses other models discussed in the research. Yet, the dataset was not pre-processed by the authors.

Table 1 displays the current models for T2D. The publications for detecting T2D ranged from 2017 to 2023, as shown in Table 1. We collected a few datasets related to T2D detection for our research. The PIDD dataset is widely utilized because it is available to the public. When working with classifiers, it's important to pre-process the dataset by eliminating unnecessary characters and handling missing values. PIDD involves incomplete and absent values. Rahman et. al. [2] utilized data pre-processing techniques to replace missing values with the median value. Yet, the majority of researchers did not pre-process the datasets. Choosing the right features plays a crucial role in determining the effectiveness of algorithms in machine learning. The data related to diabetes is utilized for training the models, which ultimately leads to producing accurate outcomes. Several researchers have employed different methods for selecting features in models to predict T2D. In [19], Feature Importance (FI) was utilized for PIDD to choose four features from a total of eight. Furthermore, the Boruta algorithm was employed in [2] for PIDD to choose five features from a total of eight. None of the previous studies have utilized rough set theory for detecting T2D.

3. The Proposed Solution

The proposed Diabetes Type Two Detection (DTTD) Rough Neuro model optimizes the combination of a neural network classifier and a rough set attribute reduction algorithm for detecting T2D. Figure 1 illustrates the proposed model, consisting of two primary phases: the rough set phase and the neural network classifier phase.

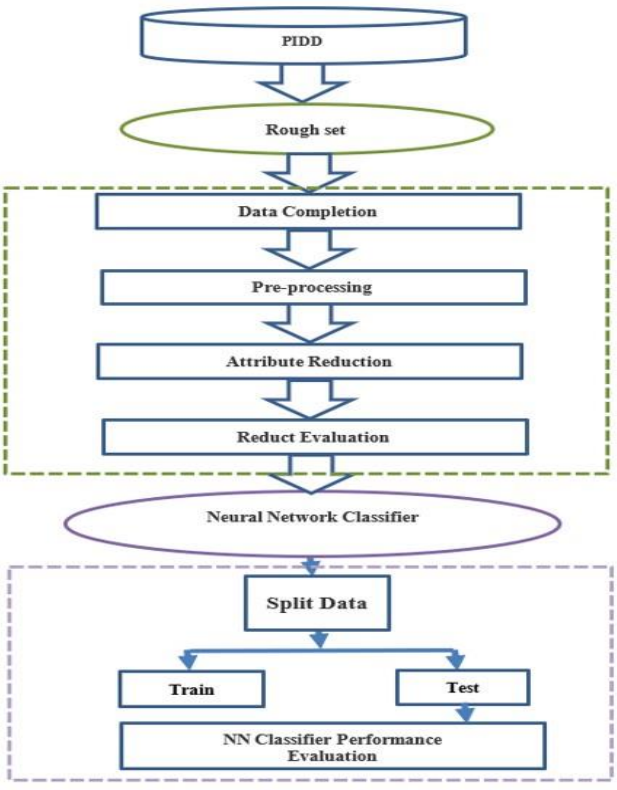


Figure. 1. DTTD Rough-Neuro Model Steps.

The dataset used in this study is the Pima Indians Diabetes Database (PIDD), which is commonly utilized and was

acquired from the University of California Irvine (UCI) machine learning repository. The dataset includes medical information for 768 individuals who are 21 years old or above, out of which 268 have been identified with diabetes. The dataset contains eight predictor variables: Pregnancy, Blood Pressure, Glucose, Skin Thickness, Body Mass Index (BMI), Diabetes Pedigree Function, Age, and Insulin. The variable of interest, referred to as "Outcome," signifies whether a patient is diagnosed with diabetes or not [2]. During the initial phases of pregnancy, blood sugar levels can increase, resulting in complications related to diabetes. An elevated blood glucose level is a crucial sign of diabetes. Elevated levels of blood sugar can increase blood pressure, which is a key indicator of diabetes. T2D tends to be more common in individuals who are overweight. Therefore, obesity greatly raises the likelihood of developing type 2 diabetes. An important indicator of diabetes is an imbalance in insulin levels. The diabetes pedigree function is valuable for obtaining diabetes information as it can be hereditary. Individuals with insulin-dependent diabetes show signs of skin thickening. The risk of T2D increases as individuals get older, particularly after the age of 45. Therefore, it can be inferred that all these characteristics are crucial for identifying T2D. The dataset's outcome column indicates a value of 1 for "tested positive for diabetes" and a value of 0 for "tested negative for diabetes". The dataset is briefly described in Table 2 [2]. PIDD contains missing values in some attributes, such as Glucose, Blood pressure, Skin Thickness, Insulin and BMI. The presence of zero in the minimum value column of these attributes means that there are missing values except for Pregnancy.

Table 1. Existing models for T2D

Year	Ref.	Dataset	Subjects	Features	Technique	Pre-processing	Feature selection	Accuracy
2023	[16]	Data collected using survey from Bangladesh's urban and rural communities	738	Many features.	DT, RF, SVM, LR and KNN	Remove outliers and filling the records with missing values	PCA and IG	82.2%
2023	[17]	PIDD	768	PC, PG, BP, ST, 2HSI, BMI, DPF, Age	KNN	-	K-means with HR, PSO and GA	91.65%
2022	[18]	PIDD	768	PC, PG, BP, ST, 2HSI, BMI, DPF, Age	MLP, DT, KNN, RF	Remove outliers, filling the missing values by the mean	Correlation, IG and PCA	79.8%
2020	[2]	PIDD	768	PC, PG, BP, ST, 2HSI, BMI, DPF, Age	Convolutional LSTM	Replace the missing values by the median value	Boruta algorithm to select 5 features out 8	97.26%
2020	[19]	PIDD	768	PC, PG, BP, ST, 2HSI, BMI, DPF, Age	DNN	Eliminate empty, redundant or any ambiguous data	Feature Importance to select 4 out 8	98.16%
2020	[20]	1-PIDD 2-DTD	1-768 2-1009	1-PC, PG,BP, ST, 2HSI, BMI, DPF, Age 2- Age, BS_fast, BS_PP, Plasma_R, Plasma_F, HbA1c, Type	DNN	Split data into training and testing data	-	1-PIDD: 99.41% 2-DTD: 94.02%
2020	[21]	PIDD	768	PC, PG, BP, ST, 2HSI, BMI, DPF, Age	ANN, NB, DT, Multilayer feed forward perceptron	Not mentioned	-	Multilayer feed forward perceptron: 98.07%
2019	[22]	PIDD	768	PC, PG, BP, ST, 2HSI, BMI, DPF, Age	Fuzzy SVM	Remove ST and 2HSI	F-score to select PG and BMI	89.02%
2019	[23]	PIDD	768	PC, PG, BP, ST, 2HSI, BMI, DPF, Age	Deep BNN	Apply normalization technique	-	80.8%
2019	[24]	PIDD	768	PC, PG, BP, ST, 2HSI, BMI, DPF, Age	MLP	Encoding the categorical data and independent variables	-	86.67%

2019	[25]	Record data sourced by Practice Fusion from public Hospitals EHRs	9948	1-Fixed and basic features 2-Adjustable features 3-Crossed features	Ensemble model	-	-	84.28%
2019	[26]	PIDD	768	PC, PG, BP, ST, 2HSI, BMI, DPF, Age	DNN	-	-	86.26%
2018	[27]	PIDD	768	PC, PG, BP, ST, 2HSI, BMI, DPF, Age	CNN	-	-	95%
2017	[28]	PIDD	768	PC, PG, BP, ST, 2HSI, BMI, DPF, Age	DNN	-	-	88.41%

DT: Decision Tree, RF: Random Forest, SVM: Support Vector Machine, LR: Logistic Regression, KNN: k-nearest neighbors, PCA: Principal Component Analysis, IG: Information Gain, PIDD: Pima Indian Diabetes Dataset, PC: Pregnancy count, PG: Plasma Glucose, BP: Blood Pressure, ST: Skin Thickness, 2HSI: 2Hour Serum Insulin, BMI: Body Mass Index, DPF: Diabetes Pedigree Function, MLP: Multilayer Perceptron, LSTM: Long Short Term Memory, DNN: Deep Neural Network, DTD: Diabetes Type Dataset, BS_fast: Fasting Blood sugar, BS_PP: Blood sugar 90 minutes post meal, Plasma_R: randomly taken Plasma glucose test, Plasma_F: Plasma glucose test typically appropriated at daybreak, HbA1c: Hemoglobin A1c test, ANN: Artificial Neural Network, NB: Naive Bayes, BNN: Belief Neural Network, EHR: Electronic Health Records, CNN: Convolutional Neural Network.

Table 2. Description of PIDD

No.	Attribute	Description	Minimum value	Maximum value
1	Pregnancy	The frequency of a partaker's Pregnancies	0	17
2	Glucose	Plasma glucose concentration 2-hour oral glucose tolerance test.	0	199
3	Blood pressure	It entails Diastolic blood pressure (blood is exerted into arteries midst the heart) (mmHg).	0	122
4	Skin Thickness	Triceps skinfold thickness (mm). It's decided by the collagen content.	0	99
5	Insulin	2-Hour serum insulin (µU/mL).	0	846
6	Body Mass Index	Body mass index (heaviness in kg/(tallness in m)^2).	0	67.1
7	Diabetes pedigree Function	An interesting attribute to diagnose diabetes.	0.078	2.42
8	Age	Participants age	21	81
9	Outcome	Diabetes class variable, Yes confirms diabetes in patients and no represents an absence of diabetes in patients.	0	1

A. Phase 1: Rough Set

1. **Data Completion:** Missing values are common in real-world data. Features in the dataset with missing values may have unfavorable consequences. Removing all features with one or more missing values is the goal of the data completion procedure. Practical data analysis frequently involves information systems or incomplete data, and methods for completing the incomplete information systems are normal in knowledge discovery and data mining through different completion methods in the pre-processing phase [29]. PIDD has 768 rows. In order to apply this step, we remove any row with a 0 value in any feature except Pregnancies since 0 has meaning. Now, PIDD has 392 subjects where there are 262 non-diabetics that represent 66.84%, and 130 diabetics that represent 33.16%.
2. **Data Pre-processing:** Cleaning the dataset is crucial before utilizing it. Following the data completion phase, the dataset is pre-processed by normalizing it with StandardScaler.
3. **Attribute Reduction:** There are frequently restricted characteristics that do not offer "almost" any further evidence about the objects. To make the decision process less complicated and less expensive, these attributes must be eliminated. However, identifying all the reducts is a complex task, but fortunately, in practical scenarios, one or a few of them are typically sufficient, and it is not always essential to identify all of them. Rough set theory

provides valuable techniques for eliminating redundant and irrelevant attributes from large datasets with numerous attributes. In rough set theory, the two common attribute reduction measures are information entropy and dependence degree, also referred to as classification quality or approximation quality [6]. Table 3 and 4 show a sample of PIDD and two of its reducts.

4. **Reduct Evaluation:** The selection of the optimal reduct is crucial and depends on attributes that meet specific optimality criteria. In this paper, three criteria are utilized [6]:

- Cardinality: This factor takes into account the quantity of attributes in the reduct. A lower number of attributes indicates a more optimal reduct. Just looking at cardinality is not enough to greatly reduce the number of reduction algorithms to choose from. As a result, more measurements are required.

- The number of rules generated is comparable to the cardinality of the reduct, highlighting the preference for fewer rules to achieve a better reduct.

- Support is calculated by dividing all objects classified by all objects to be classified, which is the dimension of training. A greater level of support indicates a more optimal reduct.

These factors are crucial signs for choosing the most suitable reduction in the research context.

Table 3. A sample of PIDD Dataset and two different reducts (Part 1).

No.	Pregnancies (a)	Glucose (b)	Blood Pressure (c)	Skin Thickness (d)	Insulin (e)	Body Mass Index (f)	Diabetes Pedigree Function (g)	Age (h)	Outcome (i)
1	6	148	72	35	0	33.6	0.627	50	1 (Diabetic)
2	1	85	66	29	0	26.6	0.351	31	0 (Non-Diabetic)
3	8	183	64	0	0	23.3	0.672	32	1 (Diabetic)
4	1	89	66	23	94	28.1	0.167	21	0 (Non-Diabetic)
5	0	137	40	35	168	43.1	2.288	33	1 (Diabetic)
6	5	116	74	0	0	25.6	0.201	30	0 (Non-Diabetic)

Table 4. A sample of PIDD dataset and two different reducts (Part 2).

No.	c	e	g	i	No.	b	f	h	I
1	72	0	0.627	1 (Diabetic)	1	148	33.6	50	1 (Diabetic)
2	66	0	0.351	0 (Non-Diabetic)	2	85	26.6	31	0 (Non-Diabetic)
3	64	0	0.672	1 (Diabetic)	3	183	23.3	32	1 (Diabetic)
4	66	94	0.167	0 (Non-Diabetic)	4	89	28.1	21	0 (Non-Diabetic)
5	40	168	2.288	1 (Diabetic)	5	137	43.1	33	1 (Diabetic)
6	74	0	0.201	0 (Non-Diabetic)	6	116	25.6	30	0 (Non-Diabetic)

B. Phase 2: Classifier using Neural Networks

To confirm the practicality of the reduction, a classifier constructed using MLP was put into action. To find the optimal setup, a series of experiments were carried out involving the network's configuration and initiation functions [6].

1. Structure of the network:

- **Network Structure:** The classifier utilized is MLP. The proposed solution's network consists of three layers: a) On the input side, this solution uses two different networks depending on the number of inputs. The initial configuration contains eight input neurons, representing the number of features before reduction. The second network has a reduced number of three inputs. b) The number of neurons in the hidden layer varies from half the inputs to twice the inputs plus one. It is important to analyze the data in various scenarios involving different numbers of hidden layers, ranging from 3 to 17 hidden neurons. This paper presents findings from experiments conducted with varying numbers of neurons in the hidden layer, ranging from 3 to 7. c) Output layer consisted of a single neuron due to the nature of the binary classification task. There are two categories, one representing non-diabetics (0) and the other representing diabetics (1).

- **Activation Function:** The proposed solution's network utilized a ReLU activation function for the hidden layer and a sigmoid for the output layer.

2. During the data completion step, it was mentioned that the PIDD dataset includes 392 subjects. In order to make network training and testing easier, we split the dataset into two sections: data from 314 subjects for training the network (80%) and data from 78 subjects for testing the network (20%).

3. Network Training: The preparation process is completed and ready for training. Network training was done in Google Colab using Python with two different learning rate (lr) settings: lr = 0.01 and lr = 0.001. In each scenario, features are trained and tested before and after reduction using the MLP classifier. The MLP classifier includes a hidden layer that contains varying numbers of neurons, ranging from 3 to 7. Training occurs three times, with each session running for 500, 1000, and 1500 epochs respectively.

4. Classifier Evaluation: After the previous steps, the proposed method is analyzed based on the accurate results from each test. The classifier with the highest accuracy will be considered the best fit for the proposed model. Additionally, in this study, the results of pre- and post-reduction will be evaluated to examine how the rough set affects neural network classifiers.

4. Results of the Experiment and Discussion

Rough sets are essential for filling in data gaps and decreasing the input size of neural networks. Consequently, it decreases the training time and storage needs of the network. Here, the DTTD Rough-Neuro Model's performance is assessed for each phase and as a complete model. Different assessment criteria are used to evaluate the effectiveness of each stage of the model being suggested.

A. Phase 1: Rough Set

In order to diminish the number of measured attributes, the Rough Set Attributes Reduction (RSAR) algorithms developed by Aleksander Øhrn ROSETTA were employed on each of the four reduction algorithms. This engendered the formation of reducts and rules for each algorithm [6]. The information can be accessed in a CSV (Comma-separated value) file format, imported into ROSETTA using Microsoft Open Database Connectivity (ODBC), and employed in conjunction with each reduction algorithm [6]. Table 5 exhibits the quantity of rules and reducts generated by each algorithm for PIDD. It demonstrates that the SAVGenetic Reducer achieved the highest number of reducts and rules, while the ManualReducer demonstrated the lowest number of reducts and rules.

Table 5. The number of generated rules and reducts of each RSAR algorithm.

No.	Reduction algorithms	No. of reducts	No. of rules
1	SAVGeneticReducer	39	29,271
2	JohnsonReducer	1	768
3	Holte1RReducer	8	1,250
4	ManualReducer	1	755

As per Table 6, the ManualReducer demonstrates the smallest quantity of rules; however, it also exhibits the lowest level of support, thus making it unsuitable for inclusion in the list of available options. After removing the ManualReducer, it was discovered that the JohnsonReducer has the fewest number of rules and the highest level of support. This discovery suggests that the JohnsonReducer is the most suitable choice for the suggested model. Referring to the information presented in Table 6, it can be observed that the JohnsonReducer presents a reduced collection of attributes (Pregnancies, Glucose, and Diabetes Pedigree Function) that produces the same outcome as the original attributes, albeit with a significant 63% reduction in complexity. Figure 2 visually illustrates the pseudocode for the JohnsonReducer.

Table 6. The evaluation measurements of reduct and rules.

No.	Reduction algorithms	No. of reducts	No. of rules	Cardinalities of reduct	Support of reduct
1	SAVGeneticReducer	39	29,271	3,4	100
2	JohnsonReducer	1	768	3	100
3	Holte1RReducer	8	1,250	1	1
4	ManualReducer	1	755	3	0

Pseudo-Code: JohnsonReducer

Let A be a "universal " set of n elements, $= {$S_1$, S_2, …. S_k} a collection of subset U forming a cover for it, and c: $\rightarrow Q^+$ a cost function. Johnson´s approximation algorithm finds a sub-collection of $ covering all the elements of U at minimal cost.

Input: C =0, T=0.
Output: T

1. Step1: let C=0, T =0.
2. Step2: while C≠U do
   1. Find S € $ such that c(S)\|S\C| is minimum.
   2. ¥ x € s, define cost(x)= C(S)\ |S\C|.
   3. C←C U S, T ←C U {S}.
3. Step3: Result = T.

Figure 2. JohnsonReducer algorithm [7].

B. Phase 2: Neural Network Classifier

The performance of the proposed model was assessed based on accuracy (ACC). The precision of a model is determined by the percentage of patients correctly identified by the models as shown in (Eq. 1) [2]:

$$ACC = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

TP (True positive) represents the sum of patients identified as positive who are truly positive. Patients classified as True Negative (TN) are those expected to be negative and indeed test negative. Identifying patients as positive when they are actually negative is known as false positive (FP). When patients who are actually positive are classified as negative, it is referred to as a false negative (FN). Such parameters are often expected to assess the model's classification accuracy [2].

This section will discuss the accuracy results of the MLP classifier with varying numbers of neurons in hidden layer, comparing two scenarios: before reduction and after reduction. There are two instances of learning rate, one being 0.01 and the other 0.001. For each scenario, the classifiers underwent training for 500, 1000, and 1500 epochs.

(i) Scenario 1 (lr=0.01):

According to Figure 3, the data before reduction indicates that the top accuracy score is 78.1%, achieved by a classifier with 3 hidden neurons trained over 1000 epochs. Results show that the highest accuracy value is 78.61, achieved by a classifier with 7 hidden neurons trained for 1000 epochs, as seen in Figure 4. The 0.51 difference between the two values was considered insignificant, indicating that while the reduction process did not enhance the result, it did help in reducing training time and storage requirements.

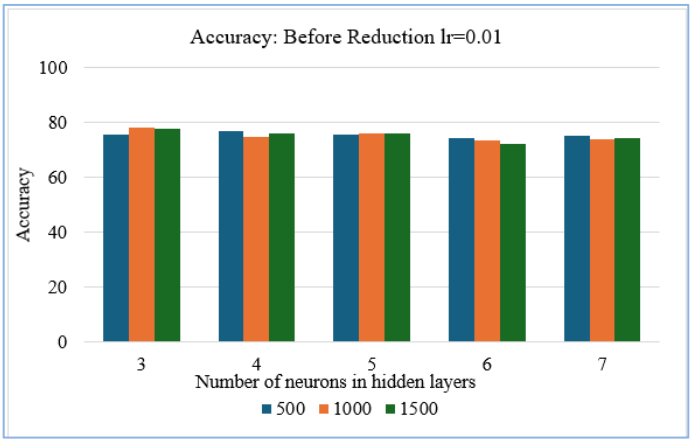


Figure 3. Before reduction accuracy results for lr=0.01.

Figure 4. After reduction accuracy results for lr=0.01.

During epoch 500, the accuracy reached 77.09 with 4 neurons in the hidden layer without reduction, whereas with reduction, the accuracy increased to 78.36 with 3 neurons in the hidden layer (refer to Figure 5). The discrepancy of 1.27 between the two values was considered insignificant, indicating that while the reduction process did not enhance the outcome, it did lead to a decrease in training time and storage requirements.

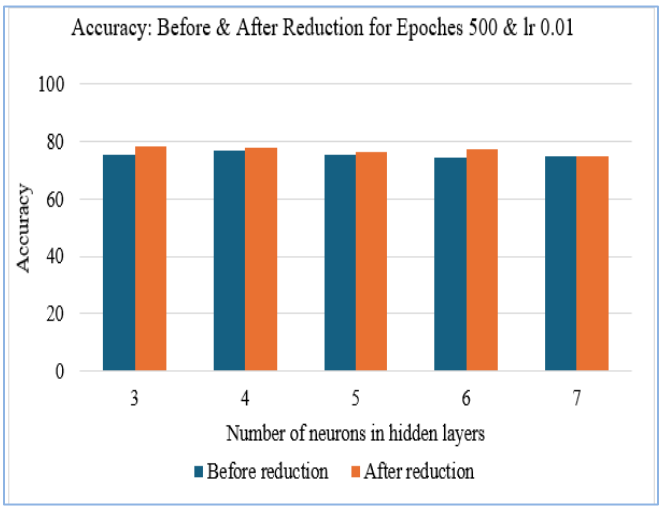


Figure 5. Results of accuracy before and after reduction for lr=0.01 and 500 epochs.

For 1000 epochs, an accuracy of 78.1 was achieved with 3 neurons in the hidden layer, while an accuracy of 78.61 was achieved with 7 neurons in the hidden layer (refer to Figure 6). The 0.51 difference between the two values was considered insignificant, indicating that the reduction process did not enhance the result, but it did lead to a decrease in training time and storage requirements.

Figure 6. Results of accuracy before and after reduction for lr=0.01 and 1000 epochs.

For 1500 epochs, the accuracy reached 77.85 without any reduction, with 3 neurons in the hidden layer. The accuracy dropped slightly to 77.09 with the reduction, still using 3 neurons in the hidden layer (refer to Figure 7). The discrepancy between the two values was 0.76, which was considered insignificant. It was observed that while the reduction process did not enhance the outcome, it did decrease both training time and storage requirements.

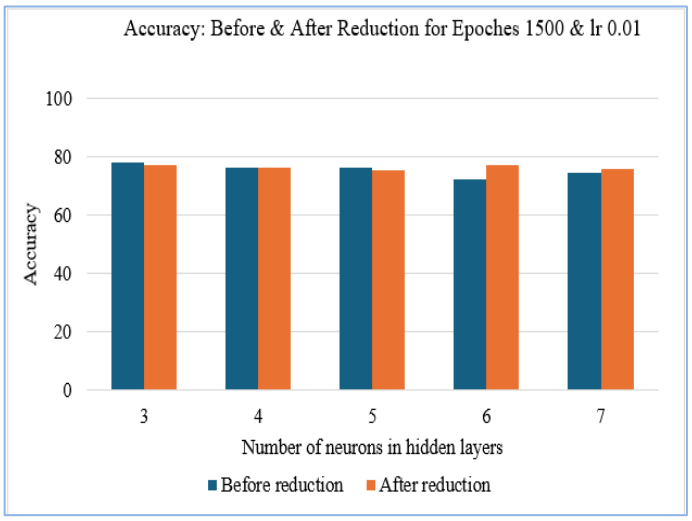


Figure 7. Results of accuracy before and after reduction for lr=0.01 and 1500 epochs.

(ii) Scenario 2 (lr=0.001):

According to Figure 8, the top accuracy achieved was 79.37% by classifiers with 3 hidden neurons trained for 500 epochs. In Figure 9, it is evident that the classifier with 3 hidden neurons trained for 1000 epochs achieved an accuracy of 78.48, the highest among all. The discrepancy of 0.89 between the two values was considered insignificant, indicating that while the reduction process did not enhance the outcome, it did lead to savings in training time and storage.

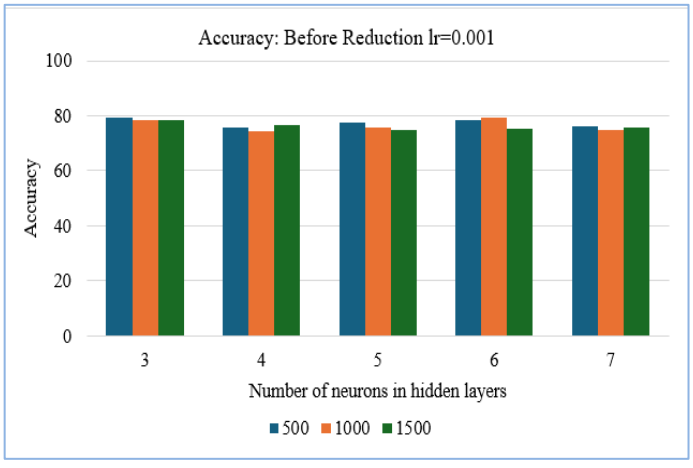


Figure 8. Accuracy outcomes before reduction for lr=0.001.

Figure 9. Accuracy outcomes after reduction for lr=0.001.

Epochs 500 saw the highest accuracy of 79.37 with 3 neurons in the hidden layer before reduction, and 77.22 with 5 neurons in the hidden layer after reduction (refer to Figure 10). The 2.15 difference between the two values was considered insignificant, indicating that while the reduction process did not enhance the result, it did help in reducing training time and storage requirements.

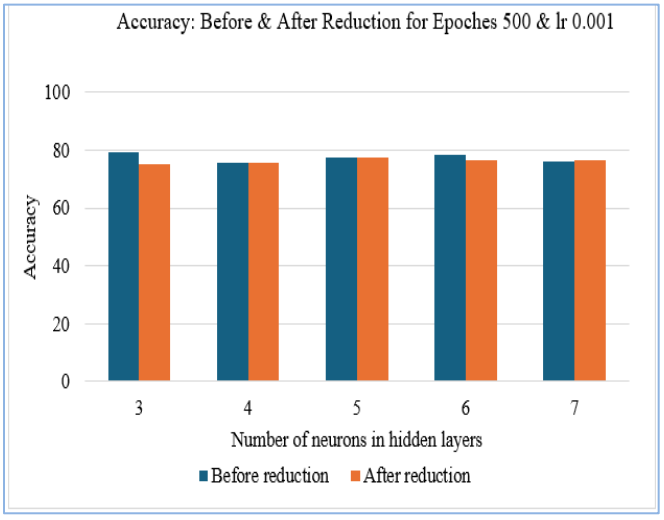


Figure 10. Results of accuracy before and after reduction for lr=0.001 and 500 epochs.

Epochs 1000 yielded an accuracy of 79.11 was achieved with 6 neurons in the hidden layer prior to reduction, and an accuracy of 78.48 was achieved with 3 neurons in the hidden layer after reduction (see Figure 11). The difference of 0.63 between these two values was deemed inconsequential. It was noted that although the reduction process did not improve the outcome, it did assist in reducing the time required for training and the storage demands.

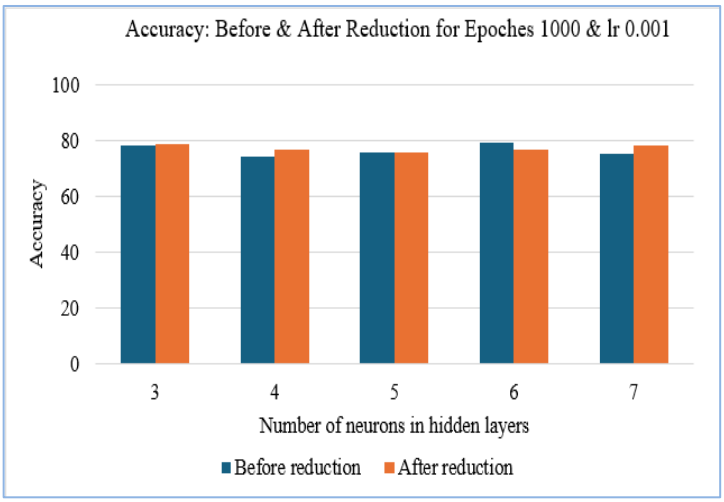


Figure 11. Results of accuracy before and after reduction for lr=0.001 and 1000 epochs.

For 1500 epochs, the scenario observed a peak accuracy of 78.23 with a hidden layer consisting of 3 neurons. In contrast, the scenario following reduction achieved a peak accuracy of 77.34 with 5 neurons in the hidden layer (see Figure 12). The insignificant difference of 0.89 between these two values suggests that while the reduction process did not improve the outcome, it did result in reduced training time and storage requirements.

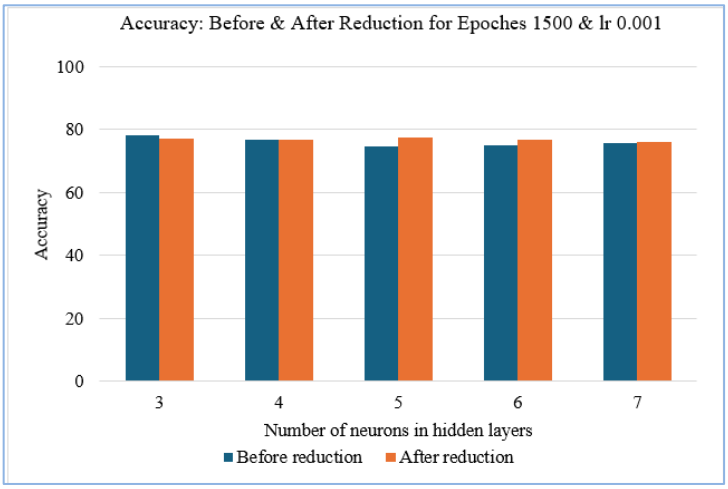


Figure 12. Results of accuracy before and after reduction for lr=0.001 and 1500 epochs.

Based on the data provided, it was noted that the greatest level of accuracy is achieved by decreasing the number of inputs. It's possible to achieve the same outcome using a smaller neural network (3 Input- 7 Hidden- 1 Output) with a learning rate of 0.01 trained over 1000 iterations, resulting in significant reductions in time and storage requirements.

There are various factors that may have contributed to the lower classification accuracy (79%) of the approach presented in Table 1 in comparison to other studies that have reported higher accuracies (above 90%) on the PIDD dataset. Consider

the following factors:

1) Dataset Variations: The specific makeup of the PIDD dataset may differ across various research projects. Variations in dataset characteristics can arise from differences in data pre-processing, and the choice to include or exclude specific features.

2) Selecting the right features for the classification model is crucial. Certain studies might have chosen a specific group of valuable characteristics or applied methods to improve predictive accuracy.

3) Model Selection: Various studies may have used more sophisticated or ensemble models that are more appropriate for the dataset's characteristics. The selection of the machine learning model and its hyperparameters can have a substantial effect on performance.

4) The dataset for PIDD may have an imbalance, where one class (such as non-diabetic) has more instances than the other (such as diabetic). Addressing the imbalance in class distribution is crucial for achieving precise classification, and different research works may have employed a range of methods to tackle this issue.

5) Cross-validation can impact the reported accuracy based on the chosen strategy, such as k-fold cross-validation. It is important for the researchers to implement a thorough and uniform cross-validation methodology to ensure precise comparisons.

6) Overfitting occurs when a model achieves extremely high accuracy by fitting the training data too closely, resulting in poor performance on unseen data. A more traditional approach might result in slightly lower accuracy, but it could lead to improved generalization.

7) Data Pre-processing: The effectiveness of data pre-processing such as managing missing values, outliers, and normalization, can impact the performance of the model. Varying pre-processing methods can result in diverse outcomes.

8) Dataset size can vary between studies when used for training and testing. Having more extensive datasets can result in the development of stronger models.

9) Publication Bias: It is important to acknowledge that publications frequently tend to only report positive and significant findings. Research with lower accuracies might not be published or featured as often.

In order to enhance the accuracy of classification, the researchers should thoroughly analyze various factors such as data pre-processing, feature selection, model selection, and hyperparameter tuning. Furthermore, they can delve into more sophisticated machine learning approaches and explore ensembling techniques to improve model accuracy. It is crucial to provide clear details about the methods and results, such as addressing class imbalance and conducting thorough cross-validation, to fully grasp the model's performance.

5. Conclusion

The study introduced a Rough-Neuro classification model using a two-stage approach to detect type 2 diabetes. The methodology utilizes rough sets from JohnsonReducer to minimize relevant attributes, while disease classification is done using a multilayer perceptron. The aim of the proposed solution is to minimize the inputs, thereby decreasing the time and storage required for training the neural network. The solution proposed is designed to reduce the input features, resulting in a reduction in both neural network training time and storage needs. The outcomes illustrate that a decrease in the quantity of input features induces a reduction in the duration of training for neural networks, an enhancement in the performance of the model, and a notable decline of 63% in the necessities for storage. These findings confirm that fewer input features result in faster training, enhanced accuracy, and reduced storage demands. Moreover, the most favorable outcomes were attained through the training of a compact neural network (3 Input - 7 Hidden - 1 Output) utilizing a learning rate of 0.01 over 1000 iterations, subsequently leading to a remarkable decline in time and storage requirements. Future improvements for the proposed solution involve training a neural network model using hybrid models. This involves exploring how various machine learning algorithms can be combined with neural networks. Blending various methods can lead to more effective outcomes. Next, disease progression modelling involves expanding the model's abilities to predict disease progression and risk factors.

Conflict of interest The authors declare no conflict of interests. All authors read and approved the final version of the paper.

Authors Contribution All authors contributed equally in this paper.

References

[1] Alfian, G., Syafrudin, M., Ijaz, M. F., Syaekhoni, M. A., Fitriyani, N. L., & Rhee, J. (2018). A personalized healthcare monitoring system for diabetic patients by utilizing BLE-based sensors and real-time data processing. Sensors, 18(7), 2183.

[2] Rahman, M., Islam, D., Mukti, R. J., & Saha, I. (2020). A deep learning approach based on convolutional LSTM for detecting diabetes. Computational Biology and Chemistry, 88, 107329.

[3] Sharma, N., & Singh, A. (2019). Diabetes detection and prediction using machine learning/IoT: A survey. In Advanced Informatics for Computing Research: Second International Conference, ICAICR 2018, Shimla, In- dia, July 14–15, 2018, Revised Selected Papers, Part I 2 (pp. 471-479). Springer Singapore.

[4] Larabi-Marie-Sainte, S., Aburahmah, L., Almohaini, R., & Saba, T. (2019). Current techniques for diabetes prediction: review and case study. Applied Sciences, 9(21), 4604.

[5] Chaki, J., Ganesh, S. T., Cidham, S. K., & Theertan, S. A. (2022). Machine learning and artificial intelligence based Diabetes Mellitus detection and self-management: A systematic review. Journal of King Saud University-Computer and Information Sciences, 34(6), 3204-3225.

[6] Rashad, M. Z. (2012). A rough-Neuro model for classifying opponent behavior in real time strategy games. AIRCC's International Journal of Computer Science and Information Technology, 185-196.

[7] Sarhan, S., Elharir, E., & Zakaria, M. (2014). A hybrid rough-neuro model for diagnosing erythemato-squamous diseases. International Journal of Computer Science Issues (IJCSI), 11(1), 143.

[8] Øhrn, A. (2000). Discernibility and rough sets in medicine: tools and applications.

[9] Slowinski, R., Zopounidis, C., & Dimitras, A. I. (1997). Prediction of company acquisition in Greece by means of the rough set approach.

European Journal of Operational Research, 100(1), 1-15.

[10] Jensen, R., & Shen, Q. (2005). Fuzzy-rough data reduction with ant colony optimization. Fuzzy sets and systems, 149(1), 5-20.

[11] Jensen, R., & Shen, Q. (2004). Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. Ieee Transactions on Knowledge and Data Engineering, 16(12), 1457-1471.

[12] Tsataltzinos, T., Iliadis, L., & Spartalis, S. (2011, September). A generalized fuzzy-rough set application for forest fire risk estimation feature reduction. In International Conference on Engineering Applications of Neural Networks (pp. 332-341). Berlin, Heidelberg: Springer Berlin Heidelberg.

[13] Karavidic´, Z., & Projovic´, D. (2018). A multi-criteria decision-making (MCDM) model in the security forces operations based on rough sets. Decision Making: Applications in Management and Engineering, 1(1), 97-120.

[14] Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. Journal of Engineering and Applied Sciences, 12(16), 4102-4107.

[15] Mukherjee, S., & Sharma, N. (2012). Intrusion detection using naive Bayes classifier with feature reduction. Procedia Technology, 4, 119-128.

[16] Kakoly, I. J., Hoque, M. R., & Hasan, N. (2023). Data-driven diabetes risk factor prediction using machine learning algorithms with feature selection technique. Sustainability, 15(6), 4930.

[17] Li, X., Zhang, J., & Safara, F. (2023). Improving the accuracy of diabetes diagnosis applications through a hybrid feature selection algorithm. Neural Processing Letters, 55(1), 153-169.

[18] Saxena, R., Sharma, S. K., Gupta, M., & Sampada, G. C. (2022). A novel approach for feature selection and classification of diabetes mellitus: machine learning methods. Computational Intelligence and Neuroscience, 2022(1), 3820360.

[19] Nadesh, R. K., & Arivuselvan, K. (2020). Type 2: diabetes mellitus prediction using deep neural networks classifier. International Journal of Cognitive Computing in Engineering, 1, 55-61.

[20] Zhou, H., Myrzashova, R., & Zheng, R. (2020). Diabetes prediction model based on an enhanced deep neural network. EURASIP Journal on Wireless Communications and Networking, 2020, 1-13.

[21] Naz, H., & Ahuja, S. (2020). Deep learning approach for diabetes prediction using PIMA Indian dataset. Journal of Diabetes & Metabolic Disorders, 19, 391-403.

[22] Lukmanto, R. B., Nugroho, A., & Akbar, H. (2019). Early detection of diabetes mellitus using feature selection and fuzzy support vector machine. Procedia Computer Science, 157, 46-54.

[23] Prabhu, P., & Selvabharathi, S. (2019, July). Deep belief neural network model for prediction of diabetes mellitus. In 2019 3rd International Conference on Imaging, Signal Processing and Communication (ICISPC) (pp. 138-142). IEEE.

[24] Muni Kumar, N., & Manjula, R. (2019). Design of multi-layer perceptron for the diagnosis of diabetes mellitus using keras in deep learning. In Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018, Volume 1 (pp. 703-711). Springer Singapore.

[25] Nguyen, B. P., Pham, H. N., Tran, H., Nghiem, N., Nguyen, Q. H., Do, T. T., ... & Simpson, C. R. (2019). Predicting the onset of type 2 diabetes using wide and deep learning with electronic health records. Computer Methods and Programs in Biomedicine, 182, 105055.

[26] Kannadasan, K., Edla, D. R., & Kuppili, V. (2019). Type 2 diabetes data classification using stacked autoencoders in deep neural networks. Clinical Epidemiology and Global Health, 7(4), 530-535.

[27] Deshmukh, T., & Fadewar, H. S. (2019). Fuzzy deep learning for diabetes detection. In Computing, Communication and Signal Processing: Proceedings of ICCASP 2018 (pp. 875-882). Springer Singapore.

[28] Ashiquzzaman, A., Tushar, A. K., Islam, M. R., Shon, D., Im, K., Park, J. H., ... & Kim, J. (2018). Reduction of overfitting in diabetes prediction using deep learning neural network. In IT Convergence and Security 2017: Volume 1 (pp. 35-43). Springer Singapore.

[29] "Freecodecamp", https://www.freecodecamp.org/news/howto-handle-missing-data-in-a-dataset. Accessed: 2023-05-09.