Timely Detection of Diabetes with Support Vector Machines, Neural Networks and Deep Neural Networks

RUMEN VALCHEV[1], MIROSLAV NIKOLOV[1], OGNYAN NAKOV[2],
MILENA LAZAROVA[2], VALERI MLADENOV[1]
[1]Department Fundamentals of Electrical Engineering,
Technical University of Sofia,
Sofia, 8 St. Kliment Ohridski Blvd.,
BULGARIA

[2]Faculty Computer Systems and Technologies,
Technical University of Sofia,
Sofia, 8 St. Kliment Ohridski Blvd.,
BULGARIA

Abstract: - In this paper, we describe an expert system with three tools - Support Vector Machine (SVM), Deep Neural Network (DNN), and feed-forward neural network (NN) in MATLAB and Python to identify potential candidates with diabetes at the initial stages of the disease. To achieve this goal, the importance of the main factors associated with previous health problems and the onset of diabetes in individuals with a medical history is analyzed. By recognizing the common early indications of diabetes, the system can aid in the selection of patients and potentially benefit them by detecting the disease at an early stage and applying appropriate and timely healing.

Key-Words: - Diabetes, Diabetes early detection Support Vector Machine (SVM), Deep Neural Network (DNN), feed-forward neural network (NN)

Received: August 4, 2022. Revised: June 21, 2023. Accepted: July 29, 2023. Published: September 7, 2023.

1 Introduction

The recruitment of potential participants for research studies, such as clinical trials, involves three main stages - identification, an approach, and obtaining their consent, [1], [2]. The enrollment of patients is the most time-consuming aspect of clinical trial processes and can cause delays in meeting deadlines, with patient recruitment taking up to 30% of the clinical timeline, [3], [4]. Typically, healthcare professionals, such as doctors and nurses, are responsible for identifying and approaching potential participants. However, screening for eligible patients can be a cumbersome and error-prone process, particularly in determining which patients meet the eligibility criteria of the respective clinical trials, [5], [6], [7]. This process is often conducted manually and is a time-consuming procedure, as each trial has specific inclusion and exclusion criteria. Additionally, even if a patient is eligible for a particular clinical trial, they may not be at the required stage of the disease specified in the protocol, while suitable patients may choose not to participate due to travel inconvenience, lack of time, out-of-pocket expenses, or simply being unaware of relevant trials, [7], [8], [9]. An experiment was carried out to identify eligible patients for a clinical trial targeting individuals in the early stages of diabetes, with a focus on detecting patients before they are formally diagnosed with the disease, [10], [11]. AI techniques, including Support Vector Machines (SVM), feed-forward neural networks (NN), deep neural networks (DNN), and k-means clustering were employed to select potential participants based on their medical histories, and the classification of diabetes was conducted using common early symptoms of the condition, [12], [13], [14]. Machine Learning (ML) is a set of many techniques, including SVM as supervised learning and k-means clustering as a non-supervised learning method.

Diabetes is a chronic and significant disease in which the level of glucose in the human blood is increased, [14], [15]. The International Diabetes Federation (IDF) reports that diabetes affects around 600,000 individuals in Bulgaria, with a higher frequency in women, and individuals between the

ages of 30 and 60, [12], [14]. As a chronic autoimmune condition, diabetes damages nerves and blood vessels, leading to complications such as heart disease, heart attack, kidney disease, and more impairments, [11]. Selecting an effective treatment is crucial to prevent disability, but the specific patient response to treatment can vary significantly. Thus, it is crucial to accurately and timely identify the individual response to medical treatment for efficient personalized diabetes therapy, [5], [7]. The progression of diabetes is unpredictable and the forecasting of the disease course in each patient would be highly beneficial for tailoring medical treatment to individual needs, [10], [12]. While several prognostic factors of disability have been identified, and some studies have proposed risk scores calculated from demographic and clinical variables collected at disease onset, [11], [13], predicting the course of diabetes based on clinical and supportive data remains challenging, and there is no currently available validated prediction model for the clinical course, [8], [9].

Numerous studies have delved into various support vector machines, neural networks, deep neural networks, and k-means clustering techniques aimed at predicting the progression and course of diabetes due to the pressing clinical need to improve the condition of diabetes patients, [12], [16], [17]. Since the natural course of diabetes can vary significantly from very mild to highly violent forms, some researchers have concentrated on creating a tailored prediction model for each diabetes patient by utilizing SVM, feed-forward neural networks, Big Data, and deep neural networks methods, [7], [8]. These studies have revealed that sizable and well-managed clinical databases can be effectively utilized to forecast the progress of diabetes in individual patients, encouraging physicians to make efforts in organizing their data in computer-friendly formats, [12], [18], [19]. Further investigation has demonstrated that the development of collaborative decision-making and visualization tools between physicians and patients can be facilitated with predictive algorithms, and that machine learning techniques can be potent tools for personalizing diabetes medical treatment strategies, [19], [20], [21].

Detecting diabetes at an early stage is crucial since it provides us with the opportunity to pursue medical treatment and plan. As the disability caused by diabetes tends to accumulate gradually, it is imperative to have an accurate and timely medical diagnosis, [11], [14]. Diabetes early detection enables medical practitioners to initiate prompt treatment, which has been found to delay the onset of disability, slow the accumulation of disability, and potentially prevent or postpone aggravations, [3], [11], [12]. The motivation for this research is a partial deficit of different techniques for the evaluation of the risk factors for diabetes progression and the recognition of potential patients in risk groups.

The main purpose of this paper is to propose an expert system, containing three basic modules – Support Vector Machine, Feed-forward Neural Network, and Deep Neural Network for early detection of diabetes in potential patients, using their medical history, taking into account the risk factors and parameters and applying different programming environments, as MATLAB, [19], and Python, [22], [23], [24], for evaluation of the main important parameters and the extent of correct recognitions, [25]. In the next chapters, the modules of the considered expert system will be described in detail. The present research concentrates on the implementation of AI for early detection of diabetes by identifying and monitoring the initial symptoms, and subsequently validating the medical diagnosis, [4], [5].

This paper is structured in the following manner. The subsequent section elaborates on the collection of data and provides a description of the medical data pre-processing. Following this, in section three, we put forth three distinct approaches for the early detection of diabetes – support vector machine, k-means clustering, and a feed-forward neural network in MATLAB and Python. In Section 4, a detailed description of the implementation of the deep neural network in Python and a comparison of the derived results are presented. In the final chapter, we present the concluding remarks.

2 Data Collection, Data Description, Data Cleaning and Normalization

Diabetes typically manifests in a variety of ways, with the more prevalent symptoms as frequent urination, strong thirst, blurred vision, weakness, easy fatigue, unexpected weight loss, feeling very hungry, dizziness, vomiting, bad breath, frequent urinary tract infections, menopause, dry and itchy skin, slow wound healing, [2], [3]. Problems noted in this disease are high blood sugar levels, increased waist size due to accumulated abdomen fat, higher blood pressure, and abnormal cholesterol, and triglyceride levels in the human blood, [1], [2]. To train the support vector machine, feed-forward neural network, and deep neural network, we utilized medical history data of patients that were

randomly generated, [3], [6]. We employed two types of training methods: supervised methods by SVM, feed-forward neural networks and deep neural networks in MATLAB and Python, and non-supervised methods by k-means clustering. In the supervised approaches, a training dataset was used to train the neural network algorithms to identify early symptoms of diabetes, monitor their progression, and develop a predictive model that could assess the likelihood of the patient developing the condition, [16]. In contrast, in the non-supervised approach, we attempted to categorize data so that patients who are at higher risk, those who are at a medium risk of diabetes, and the humans who are not at risk of diabetes were sorted into three different clusters, [17]. The dataset applied in the present evaluation was obtained from Kiowa County Memorial Hospital and includes 13,850 respondents who did not provide any personal information.

From this medical dataset, we selected 25 factors related to diabetes disease and one outcome variable, DIABETESN, which indicates whether a patient is prediabetes, diabetic, or does not have diabetes. The extracted data often suffer from incompleteness, ambiguity, noise, and other difficulties that could negatively impact the performance of the respective predictive models, [19], [20], [22]. Therefore, preprocessing and validation are necessary to avoid potential issues. All the input data are normalized, applying a division of each numerical value of a given parameter by its maximal value. In this way, the numerical values of input signals are in the range between zero and unity. Statistical processing with removing the outliers is also applied. Extracting variables from patient data can be a time-consuming and laborious process, but statistical analysis software was successfully employed to extract the necessary variables from the raw data and to format the data in a manner suitable for our research purposes. To handle missing patient IDs, the corresponding clinical records were removed. The data was then subset to include the most commonly reported terms for the medical history among diabetes, non-diabetes patients, and medium-diabetes patients. The absence of diabetes is coded with 1, the pre-diabetic state – with 2, and with 3 we code the presence of diabetes. For the SVM, feed-forward neural network, and Deep Neural Network, used in this research, the data for analysis consists of the fifteen most commonly reported Medical History terms for diabetes patients in the respective raw data: 'Frequent urination, strong thirst, blurred vision, weakness, easy fatigue, unexpected weight loss, feeling very hungry, dizziness, vomiting, bad breath, frequent urinary tract infections, menopause, dry and itchy skin, slow wound healing', [12], [13]. These terms were used as features, grouped in vectors, and the derived feature vector for each patient consisted of 25 features. The parameters and their meaning are described in detail in the next section. The used features are divided into two main groups - quantitative and qualitative parameters, respectively.

3 Application of Support Vector Machine, K-Means Clustering, and a Feed-Forward Neural Network for Early Diabetes Detection

The focus of our study is to investigate how AI techniques such as Support Vector Machine and a feed-forward neural network could be utilized to automatically analyze databases containing clinical trial eligibility criteria and electronic health records (EHRs), [1], [2]. Our main goal is to identify suitable patients for ongoing clinical trials and recommend these matches to both patients and investigators based on specific medical diagnoses, [4]. To achieve this objective, we employ the commonly used techniques, namely SVM, neural network in MATLAB, and DNN, to study the collected medical datasets. By utilizing these techniques, we aim to detect diabetes at an early stage and thereby improve the recruitment process for clinical trials. In the next subsection, the parameters applied for early detection of diabetes are described, for a better understanding of the following explanations and the related study. The accuracy and the respective error are calculated as the ratio of correctly and falsely recognized patients to their actual number.

3.1 Parameters Used for Early Diabetes Detection

At the beginning of the present study, the complete group of 26 parameters is applied – 1 output и 25 input parameters, respectively. The Output parameter is DIABETESN, it is coded by 1 for the absence of diabetes, with 2 for a pre-diabetic state, and with 3 for hard forms of diabetes, respectively.

The input parameters applied for the classifications are: GENDERN (the gender of the respective patient), AGE (the patients' age), ETHNICN (belonging to a certain nationality), INC_THIRSTN (the extent of thirst), FREQ_URINN (rate and amount of urination), INC_HUNGERN (an extent of human's hunger),

WGHT_LOSSN (the loss of body weight), FATIGUEN (the extent of fatigue of a certain human), BLUR_VISIONN (the respective visual impairment and blurriness), SLOW_HEALINGN (decreased rate of regeneration and health healing), FREQ_INFECTIONSN (occurrence of frequent infections), HBA1C (Average Blood Sugar Level Over 2-3M - Numeric evaluation), FASTING_SUGAR (Fasting Blood Sugar Test), GLUCOSE_TOLERANCE (Glucose Tolerance Test at 2H), RANDOM_SUGAR (Random Blood Sugar Test), DIASTOLIC (Diastolic Blood Pressure), SYSTOLIC (Systolic Blood Pressure), LDL_CHOL (LDL Cholesterol), HDL_CHOL (HDL Cholesterol), TRIGLYCERIDES (the presence of Triglycerides), ALBUMIN (Albumin (Urine Test)), NERVE_VELOCITY (Nerve conduction velocity), BMI (Body Mass Index), GUMLINE (gumline test), GLUCOSE_SCREENING (glucose level).

The accuracy that should be reached is previously defined as a parameter and applied before configuring and starting the training process of the neural network and SVM algorithms. The accuracy of the results is calculated based on 90 % of training data samples and 10 % of the dataset for testing the neural networks. The dataset for testing the neural networks is different from the respective data used for training the neural network and SVM algorithms. After testing processes of the neural network and SVM algorithms, it is established that the derived accuracy is a little bit lower than those obtained during the training procedures because the networks are trained with data that are different from those used for the testing procedures. After finishing the analyses, many additional experiments are made, using a lot of combinations and variants of the applied input parameters, and each of them is associated with reducing the number of the input parameters and features, using medical expert knowledge and recommendations, and measuring the obtained error and the corresponding accuracy of the results and the percentage of correct and incorrect recognitions. The main purpose of this procedure is to evaluate the importance of the respective input parameters and to attempt to reject some of the factors with lower significance on the final results and their precision, [17]. It is established that by reducing the number of the input parameters to less than 13, the accuracy of the obtained results and the extent of recognition is rapidly decreased. Owing to this fact, an additional experiment with 19 input parameters is conducted, and the final experiments are made using the number of input 13 parameters – 1 output and 12 input parameters, respectively. The output parameter is again the DIABETESN, having three possible values – 1, 2, and 3, related respectively to a lack of diabetes, medium diabetes, and hard diabetes forms. The reduced set of input parameters is established as AGE, INC_THIRSTN, FREQ_URINN, HBA1C, FASTING_SUGAR, GLUCOSE_TOLERANCE, RANDOM_SUGAR, LDL_CHOL, HDL_CHOL, ALBUMIN, BMI, GLUCOSE_SCREENING. The selected parameters and their meaning are represented above. The input data are organized in a matrix, each column corresponds to a given parameter, and each row is related to a certain patient under analysis. For the application of SVM, feed-forward neural network, and DNN, the rough medical data are pre-processed by scaling and normalization, because these algorithms are very sensitive to the different nature and ranges of the physical input data and they must be normalized by a division of each datum by it respective maximal value for each column of the matrix, so that the input variables for the neural networks are in the region between zero and unity, [16], [19].

3.2 SVM with All the Features (25 Parameters)

Support Vector Machine (SVM) is an algorithm in supervised learning utilized for solving problems, applying classification and regression. Its basic principle of operation involves identifying a hyperplane that optimally separates two classes of input data. The primary goal of SVM is to locate the most advantageous hyperplane for efficient partitioning of the input data. This procedure is conducted by selecting the hyperplane with the greatest distance (known as a margin) to the nearest data points belonging to the considered classes for the separation of the objects. These data points, referred to as support vectors, play a significant role in deriving the optimal hyperplane. SVM is a versatile classification technique and it is capable of handling both linear and non-linear input data spaces, applying diverse kernel functions. The kernels transform the respective data space, enabling SVM to operate within a higher-dimensional space, where finding an optimal separating hyperplane is a more easily feasible process. It is important to be noted that SVM has proficiency in complex classification tasks and could effectively handle datasets with a limited number of training examples and data. Additionally, SVM is associated and supported by its mathematically rigorous theory, which ensures its effectiveness when suitable hyper-parameters are

related and chosen. SVM is related to robust algorithms for classification and regression, very useful in data separation in linear and non-linear spaces and tasks. A supervised learning technique is utilized to analyze data for classification and regression analysis. In our case of three-class for patients' classification, 1 is assigned to a human state without diabetes, 2 is for the pre-diabetic state, and 3 corresponds to a hard form of diabetes. To train the systems – SVM, feed-forward neural network, and DNN, the medical dataset is split into two main subsets, consisting of 12465 patients (90 % of the whole data) for training, and 1385 patients (10 % of the data) for testing the neural networks (in the present case, the training to testing data ratio is 90 % to 10 %). For obtaining the results we use MATLAB software environment, [19], and Python programming language, [20], [22], [26], [27]. Figure 1 illustrates the outcomes obtained from the training process of SVM and ML in Python. Here are 25 input features and one output. The blue and green lines presented in Figure 1 correspond to the training and validation output data during training, respectively, and the red and orange lines correspond to the losses for the training and testing data. As the validation output follows the training accuracy improvement, i.e. we do not see a drop in accuracy with the validation data, we can conclude that we do not observe overfitting during the conducted experiments.

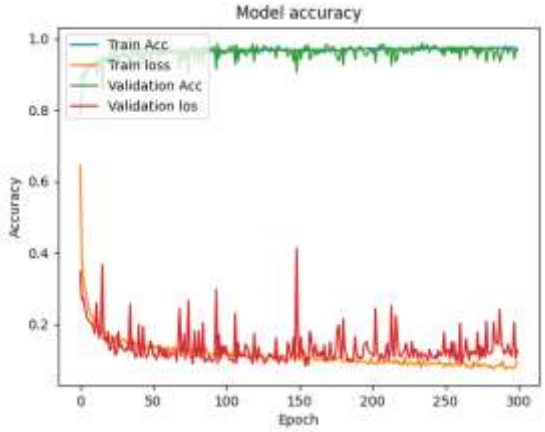

Fig. 1: Learning results and accuracy for diabetes detection by SVM with supervised learning with all 25 input parameters in Python software; loss: 0.1832 - Accuracy: 0.9617

We can observe that the accuracy of the validation output data is not very noisy, therefore the batch_size value is not too small (if the data we are operating with is noisy, we would use a larger batch_size file), also this is an indication that our optimizer is operating relatively well. Validation losses are on average smaller than training losses. This means that the training procedures were closer to the correct answer for the validation data than for the training data. Before training the systems for classifications, we set the respective accuracy to be reached and how long to continue the learning processes. The respective parameters in Python are: early_stopping_callback = EarlyStopping (monitor='val_accuracy', mode='max', patience=600, verbose=1, baseline=0.98, restore_best_weights = True). We pass as the instance of this class when we start the training processes. During these procedures, we monitor the accuracy of the respective predictive model and its best result until new epochs are launched without obtaining an improvement in accuracy (if a certain percentage of accuracy (in this case 98%) is not reached and no improvement is seen, then a break occurs of model's training). With the last parameter, we indicate the restoration of the synaptic weights of the model where the greatest accuracy is observed. An "evaluate" function gives the losses and the corresponding accuracy of the respective neural model with the testing data. The value of test_loss gives the average information about exact and correct output and test_acc is between 0 and 1 and gives the derived accuracy. The supervised learning process is conducted at about 300 epochs. After each full training session, a graph is created that shows what the performance was for each epoch. During training, the Python console displays what the prediction accuracy was for the validation data, which is five percent of the test data. The validation data is only indicative and is not fed to train the neural network. We also keep track of how accurate we get from this validation data for each epoch, and after training is complete, we take the weights of the neurons from the epoch in which we got the highest score for correctly predicting the validation data. The results from testing the system with the whole set of 25 parameters are loss: 0.1832, accuracy: 0.9617. In the next sub-section, the experiments are made and described with a reduced number of input parameters.

3.3 SVM with a Reduced Number of the Applied Features (19 Parameters)

Follows the analysis using SVM and reduced number of factors – 19 input parameters in this case, which are: Output parameter: DIABETESN; Input parameters: DIABETESN, GENDERN, AGE, ETHNICN, INC_THIRSTN, FREQ_URINN, INC_HUNGERN, WGHT_LOSSN, BLUR_VISIONN, HBA1C, FASTING_SUGAR, GLUCOSE_TOLERANCE, RANDOM_SUGAR,

DIASTOLIC, SYSTOLIC, LDL_CHOL, HDL_CHOL, ALBUMIN, BMI. The results for the losses and accuracy after testing with SVM are loss: 0.1143, accuracy: 0.9776. The results for the accuracy of diabetes detection by SVM with supervised learning with 19 input parameters are presented in Figure 2 for their observation and evaluation.

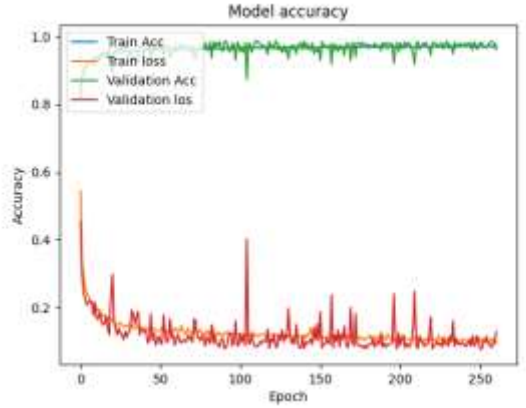

Fig. 2: Learning results and accuracy for diabetes detection by SVM with supervised learning with 19 input parameters and one output in Python; loss: 0.1143 - Accuracy: 0.9776

3.4 SVM with a Reduced Number of Applied Features (13 Input Parameters)

The results obtained using 13 input parameters are presented in Figure 3 for further descriptions.

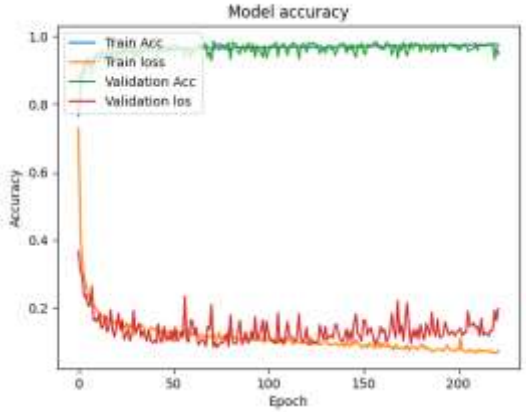

Fig. 3: Learning results for diabetes detection and accuracy by SVM with supervised learning with a reduced number of input features, here are used 13 input features in Python; loss: 0.1291 - Accuracy: 0.9769

In this case, the derived results are with precision, very near to those obtained using 19 features described in the previous subsection and applying all the input 25 parameters. The results obtained from the testing with 13 parameters are loss: 0.1291, accuracy: 0.9769.

3.5 k-means Clustering

In unsupervised learning, the main goal is to discover underlying patterns in data without previous knowledge of the correct results. K-means clustering is a part of Machine Learning (ML). Clustering analysis is a technique applied to identify natural groupings in data, where items in the same cluster have greater similarity compared to those in the other clusters. In the k-means clustering, each cluster is represented by a "prototype" data point. Based on the mean values of the respective clusters, the most appropriate number of clusters, in this case, is determined to be 3.

This is related to the actual number of classes to which the patients belong, which in this case is 3 – pre-diabetic, diabetics, and hard-diabetes, [14]. The third class is related to patients exhibiting symptoms of diabetes, while the first class represents patients without diabetes, and the second class contains patients with prediabetes symptoms, [14]. The ML analysis is realized in MATLAB and Python environments, [19]. During the k-means clustering experiments, the percentage of correct recognitions is about 50 %, and owing to this it is concluded that this method is not appropriate in the present case of analyses.

3.6 A Feed-Forward Neural Network

An artificial neural network is a hardware or software realization of an optimization algorithm for pattern recognition, time series analysis, and prediction, based on adders of the input signals, synapses with determined in the beginning synaptic weights, connected to the input nodes for applying the input signals, and each artificial neuron finishes with an activation function, which output is attached to the output node. The neural network contains a hidden layer – one more, and an output layer containing neurons. The activation (transfer) function of the neurons from the hidden layers is usually a tangent-sigmoidal or logarithmic-sigmoidal one, while the neurons from the output layers are with linear activation functions, [19]. Each layer of a neural network consists of several input nodes. The connections between the nodes and the neighboring layers of an artificial neural network are realized by synapses and depict the flow of information from a given layer to the next neural layers. A feed-forward neural network in MATLAB is applied for recognition of the state of the analyzed data. The network has 25, 19, and 13 inputs for the considered three cases of included

features. The network contains two hidden layers with 20 and 10 neurons, respectively, and a neuron with a linear activation function in the output layer.

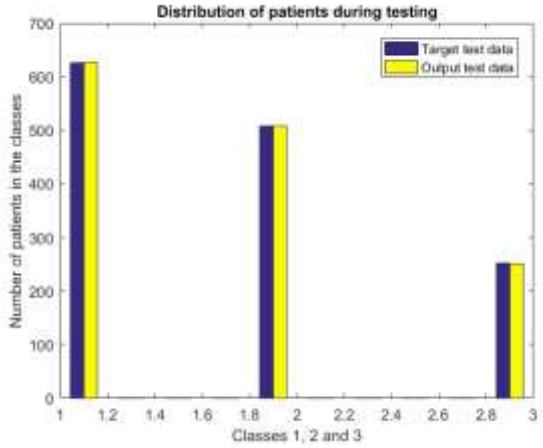

Fig. 4: Number of patents divided into three classes after testing procedures of a feed-forward neural network in MATLAB with 25 features, 20 neurons in the first hidden layer, 10 neurons in the second hidden layer

We ran multiple experiments with different numbers of features, so the respective accuracy was reported for each experiment, and its derivation from the dataset is used for testing the system. It turns out that at 13 in the number of features, the accuracy is almost the same as those obtained using 25 features.

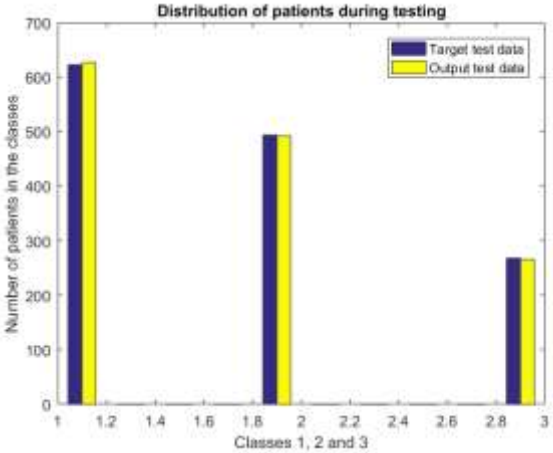

Fig. 5: Number of patents divided into three classes after testing procedures of a feed-forward neural network in MATLAB with 19 features, 20 neurons in the first hidden layer, 10 neurons in the second hidden layer

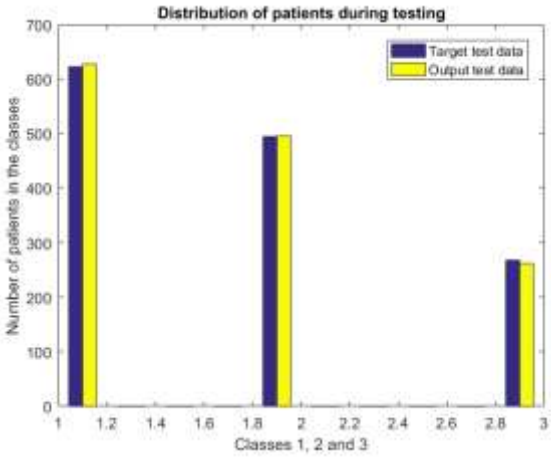

Fig. 6: Number of patents divided into three classes after testing procedures of a feed-forward neural network in MATLAB with 13 features, 20 neurons in the first hidden layer, 10 neurons in the second hidden layer

When we use several features less than 13 and the dataset for testing, we observe a rapid degradation of the recognition accuracy. The respective results are presented in Figure 4, Figure 5 and Figure 6 for their comparison and confirmation of the good efficiency of the neural network.

4 Implementation of a Deep Neural Network in Python for Early Diabetes Detection

Additional experiments are made using Deep Neural Networks in Python software, [27]. Deep learning is a branch of machine learning that deals with algorithms inspired by the structure and function of the human brain, [16], [18], [19]. It uses artificial neural networks to build intelligent models and solve complex problems. We mostly use deep learning with previously scaled and normalized data, because in many cases the neural networks are quite sensitive to the different ranges and values of the physically measured and derived input data. In Figure 7, the variable y_train in for deep learning is presented. In Figure 8, the normalized and pre-processed data for the deep neural network are represented. The learning processes of a neural network are based on a comparison between the actual derived output signal and the desired output signal, which is previously determined. The error signal of a neural network is a difference between these output signals, and it is used for adjustment of the synaptic weights of the connecting synaptic bonds. In the next section, the implementation of DNN in Python is described.

Fig. 7: A representation of the variable *y_train* in Python for deep learning

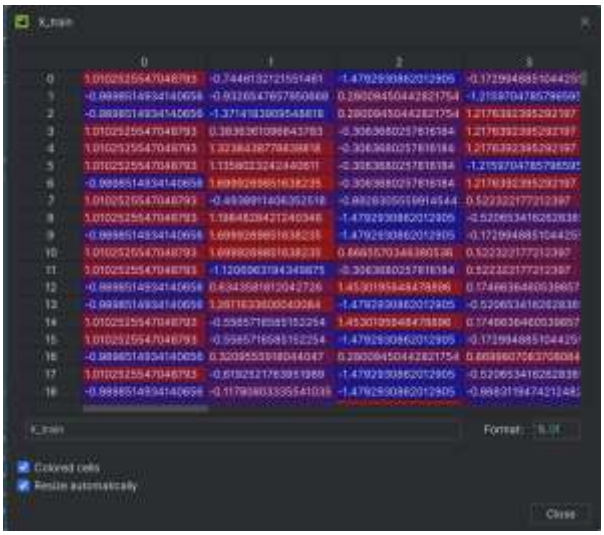

Fig. 8: Scaled data for early diabetes detection in Python for deep learning

In recent years, Python is a very frequently used and preferred software environment for artificial intelligence applications and analyses, such as Support Vector Machines, Deep Neural Networks, and others, [21], [27]. When splitting the input data, there is a slight difference in that we categorize the target with the to_categorical parameter. The variable y_train takes the following form for easier execution of the algorithm. Because we have a large set of different data sizes and ranges, and the results derived by the neural network are very sensitive to the different variations and values of the input physically recorded data, we previously scale them by divisions of their values by the respective maximal element in a given data. The keras library for the Python environment was used in this research to create the respective model for analysis of the normalized input data, [19]. The applied deep artificial neural network is of a sequential type, [21]. Four layers of the deep neural network are created, with 44, 32, 25, and 3 units, respectively. The first layer of the deep neural network is assigned to 25 inputs indicating the number of input features of each analyzed patient. The first 3 elements are set to linear rectifier units with a small slope for negative values, instead of a flat slope, and the last is based on softmax. The component softmax converts a vector of the input values into a probability distributed signal, [22]. The elements of the output vector of the deep neural network are in the range (0, 1) and their sum is unity. Softmax is often used as an activation element for the last layer of a classification deep neural network because the derived result could be interpreted as a signal with a probability distribution. A dropout is applied to avoid overfitting, [23].

Follows the description of the compiling of the model for analysis of the input data. Categorical_crossentropy parameter is suitable and applied for binary (0 or 1) classification of objects and patterns. The obtained accuracy of the results after a comparison with the desired output signal is used as a measure of quality of the recognition process, [22], [23]. The Adadelta parameter was selected as a criterion and an appropriate parameter for the optimization of the recognition and prediction processes. The early_stop parameter is used to stop the training processes of the deep neural network if improvements in the obtained results and their accuracy are not noticed, [23]. When training the used deep neural network, 20% of the initial data are used for its self-improvement. About 500 epochs are set for obtaining a better quality of the derived results. Applying the described deep neural network, two graphs are derived to visualize the used neural model for data analysis.

The graph depicted in Figure 9 is given for representation of the accuracy of the trained model of the deep neural network against the data used for its training and development while it is being trained. We could observe an increase in the quality of the derived results compared to the training data and a decrease compared to the validation data, which indicates that the deep neural network model is starting to apply to the training data (overfitting). The results from testing the system with the whole set of 26 parameters are loss: 0.5003, accuracy: 0.9486. The obtained accuracy is the result of a comparison of 3850 data of patients who did not participate in the training of the considered model.

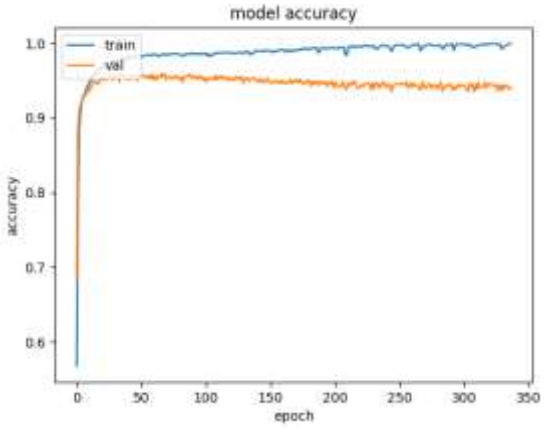

Fig. 9: Representation of the accuracy of the trained model with deep neural network against the data used for its training in Python using 26 features; 'accuracy' 0.9486

The loss of accuracy of the considered model of the deep neural network in Python is represented in Figure 10 for further comments and discussion of its operation and recognition processes. Observing Figure 10, which represents the loss of accuracy of DNN, it is clear that with increasing the number of epochs, the losses during the training processes decrease, while during the validation processes in Python, the respective losses increase.

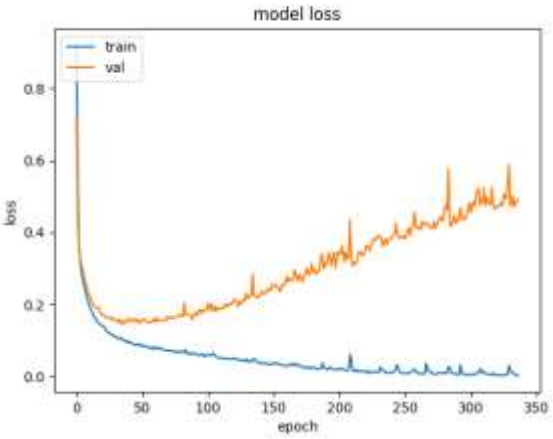

Fig. 10: A graph representing the loss of accuracy as the model is trained in a Python environment with a deep neural network using 26 features; 'loss' 0.5003

A graph for the loss of accuracy as the model is trained with a deep neural network using 19 features is presented in Figure 11. The results from testing the system with the whole set of 26 parameters are loss: 0.3924, accuracy: 0.9532.

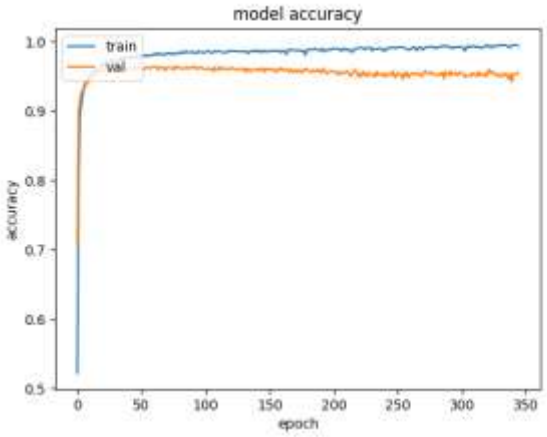

Fig. 11: A graph representing the loss of accuracy as the model is trained in a Python environment with a deep neural network using 19 features; 'accuracy' 0.9532

The accuracy of the trained model with DNN against the data for its training in Python using 19 features is shown in Figure 12.

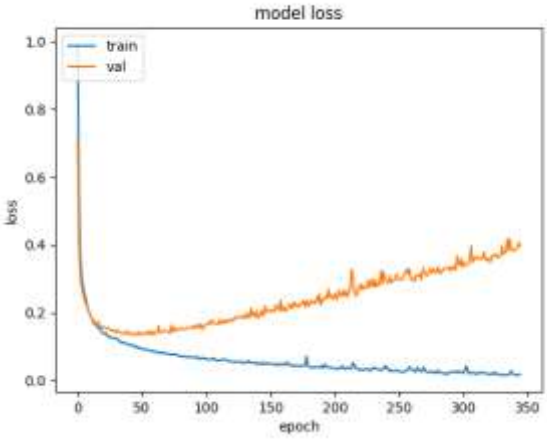

Fig. 12: Representation of the accuracy of the trained model with deep neural network against the data used for its training in Python using 19 features; 'loss', 0.3924

The respective model accuracy and losses derived by the use of 13 features are presented in Figure 13 and Figure 14. It is observable from Figure 13 that the accuracy increases with the number of epochs, while the losses presented in Figure 14 decrease during the training epochs. These figures are placed for visualization and comparison of the obtained results, using different numbers of features. The results from testing the system with the whole set of 26 parameters are loss: 0.1851, accuracy: 0.9699.

5 Results and Discussion

The results are obtained based on the three methods – SVM, NN, and DNN, used in the expert system. The proposed expert system is based on the 13 risk factors for the development of diabetes, described in section 3, owing to their highest importance and the derivation of almost the same high accuracy of the outcomes. In our case, when the derived results are very close to each other and the difference between them is about 2 – 3 %, then the results obtained with two of the methods with an exact matching are taken as outcomes, i.e., as in the election, using the dominant results. The obtained accuracy of the results is about 94.9 % after 121 epochs. It could be concluded that SVM, NN, and DNN successfully classify the potential patients under analysis into three groups – non-diabetic, pre-diabetic, and hard diabetic ones.

In Table 1, the results for accuracy derived by the applied expert system with SVM, DNN, and NN modules are summarized for comparison of these methods.

Table 1. Comparison of the accuracy of SVM, DNN, and NN, using 25, 19, and 13 features

Accuracy, %	25 features	19 features	13 features
SVM	96.2	97,8	97,7
DNN	94,9	95,3	97,0
NN	98,7	98,4	98,2

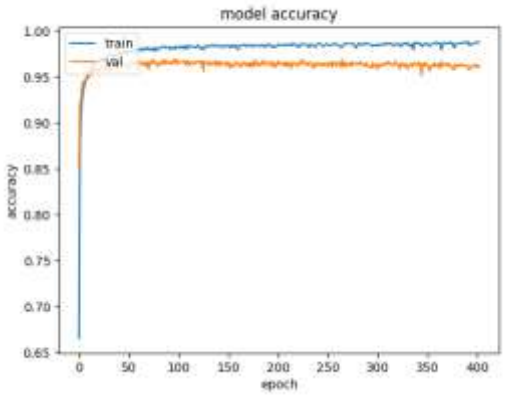

Fig. 13: Representation of the accuracy of the trained model with deep neural network against the data used for its training in Python using 13 features; 'accuracy' 0.9699

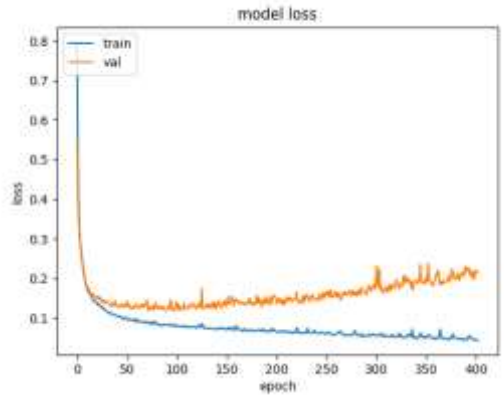

Fig. 14: Representation of the accuracy of the trained model with deep neural network against the data used for its training in Python using 13 features; 'loss', 0.1851

It is obvious that with decreasing the number of the used features, a very little decrease of accuracy is observed, but retaining the high rate of correct recognition. In some cases, a little bit higher accuracy is derived, using a decreased number of features. This outcome comes from the unavoidable noise in the input data and also due to the different importance of the features under analysis on the outcomes. The very little difference in the accuracy in such cases is in the range of the errors, related to the measurements of the features.

The main contributions in this paper are the way of reducing the applied features for the analyses of diabetes and the evaluation of their importance, so that the obtained accuracy remains high and satisfactory, and the use of an expert system for analysis of the data. In many publications related to diabetes and the main risk factors for its occurrence and development, single methods, using support vector machines, classical neural networks, or deep neural networks, are applied. In the proposed paper, an expert system with these methods is applied, and after obtaining the results, a comparison is conducted, and the dominant outcome is elected as a trusted and final result.

6 Conclusion

A detailed study of the risk factors of potentially ill patients with diabetes is conducted, using data of medical history. Based on this an expert system with three modules – SVM, NN, and Deep neural networks for recognition of three basic groups of humans – without diabetes, in a pre-diabetic state, and with a hard form of diabetes is developed. The analysis of the derived results represents a very good recognition of the potential patients with

diabetes disease – about 98 % correctly recognized humans under analysis. K-means clustering does not ensure good accuracy in this case, probably due to not very suitable scaling of the input data. Deep learning neural networks need significantly higher computing power and different experiments with increasing the number of deep neural network levels to achieve better results of pattern recognition. Overfitting was observed when training the neural network with the respective data and this suggests that this is the corresponding maximum allowable accuracy of a model of the considered type. Sometimes, it is possible that different approaches and neural network structures could lead to different and better results. The results obtained from applying SVM methods, NN, and deep neural networks to early detection of diabetes not only aid researchers in increasing the selection of suitable patients for clinical trials but also prove to be beneficial for patients, as it enables them to be diagnosed with the disease at an early stage and commence treatment promptly.

Early recognition and accurate diagnosis of diabetes are crucial in delaying disease progression and improving patient outcomes. The outcomes obtained in the present research demonstrate that the used expert system, containing SVM, NN, and DNN techniques on patients' medical history data can expedite the process of recruiting patients for clinical trials and assist in identifying patients who meet the criteria for a specific care and study.

For future work on this topic, additional analyses on different medical datasets could be conducted for the identification of the main crucial features for the prediction of diabetes development. For the determination of the most accurate and appropriate early diabetes prediction algorithms, different techniques and combinations of methods and algorithms could be analyzed.

References:

[1] Bachenheimer, J., Brescia, B., Reinventing Patient Recruitment: Revolutionary Ideas for Clinical Trial Success, *Gower Publishing*. ISBN 978-0-566-08717-2, 2007.

[2] Mujumdar, A. and Vaidehi, V., Diabetes prediction using machine learning algorithms, *Procedia Computer Science*, Vol. 165, 2019, pp. 292-299.

[3] Bhat, S.S., Selvam, V., Ansari, G.A., Ansari, M.D. and Rahman, M.H., Prevalence and early prediction of diabetes using machine learning in North Kashmir: a case study of district bandipora, *Computational Intelligence and Neuroscience*, 2022, pp. 1 - 12.

[4] Fregoso-Aparicio, L., Noguez, J., Montesinos, L. and García-García, J.A., Machine learning and deep learning predictive models for type 2 diabetes: a systematic review, *Diabetology & Metabolic Syndrome*, vol. 13, No. 1, 2021, pp.1-22.

[5] Qin, Y., Wu, J., Xiao, W., Wang, K., Huang, A., Liu, B., Yu, J., Li, C., Yu, F. and Ren, Z., Machine Learning Models for Data-Driven Prediction of Diabetes by Lifestyle Type, *International Journal of Environmental Research and Public Health*, vol. 19, No. 22, 2022, p.15027.

[6] Firdous, S., Wagai, G.A. and Sharma, K., A survey on diabetes risk prediction using machine learning approaches, *Journal of Family Medicine and Primary Care*, vol. 11, No. 11, 2022, pp.6929-6934.

[7] Temurtas, H., Yumusak, N. and Temurtas, F., A comparative study on diabetes disease diagnosis using neural networks, *Expert Systems with applications*, vol. 36, No. 4, 2009, pp. 8610-8615.

[8] Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I. and Chouvarda, I., Machine learning and data mining methods in diabetes research, *Computational and structural biotechnology journal*, vol. 15, 2017, pp. 104-116.

[9] Konasani, V.R. and Kadre, S., Machine learning and deep learning using python and tensorflow, *McGraw-Hill Education*, ISBN 9781260462296, 2021.

[10] Lee, K.D., Python programming fundamentals, London, Springer, 2011.

[11] Jordan, M.I. and Mitchell, T.M., Machine learning: Dey, S.K., Hossain, A. and Rahman, M.M., Implementation of a web application to predict diabetes disease: an approach using machine learning algorithm. *In 2018 21st IEEE international conference of computer and information technology (ICCIT)*, 2018, pp. 1-5.

[12] Sowah, R.A., Bampoe-Addo, A.A., Armoo, S.K., Saalia, F.K., Gatsi, F. and Sarkodie-Mensah, B., Design and development of diabetes management system using machine learning, *International journal of telemedicine and applications*, vol. 2020, 2020.

[13] Zhou, H., Myrzashova, R. and Zheng, R., Diabetes prediction model based on an enhanced deep neural network. *EURASIP*

Journal on Wireless Communications and Networking, 2020, pp.1-13.

[14] Pivari, F., Mingione, A., Brasacchio, C. and Soldati, L., Curcumin and type 2 diabetes mellitus: prevention and treatment. *Nutrients*, vol. 11, No. 8, p.1837, 2019.

[15] Singla, R., Singla, A., Gupta, Y. and Kalra, S., Artificial intelligence/machine learning in diabetes care. *Indian journal of endocrinology and metabolism*, vol. 23, No. 4, 2019, p.495.

[16] Pour, A.M., Seyedarabi, H., Jahromi, S.H.A. and Javadzadeh, A., Automatic detection and monitoring of diabetic retinopathy using efficient convolutional neural networks and contrast limited adaptive histogram equalization. *IEEE Access*, vol. 8, 2020, pp.136668-136673.

[17] Alić, B., Gurbeta, L. and Badnjević, A., Machine learning techniques for classification of diabetes and cardiovascular diseases. *In 2017 IEEE 6th mediterranean conference on embedded computing (MECO)* 2017, pp. 1-4.

[18] Manaswi, N.K., Manaswi, N.K. and John, S., Deep learning with applications using Python, Berkeley, CA, USA: *Apress*, ISBN-13 978-1-4842-3516-4, 2018, pp. 31-43.

[19] Raschka, S., Python machine learning. *Packt publishing ltd*. ISBN 978-1-78355-513-0, 2015.

[20] Aggarwal, C.C., Neural networks and deep learning. *Springer*, ISBN 978-3-030-06856-1, vol. 10, No. 978, 2018, p.3.

[21] Kim, P., Matlab deep learning with machine learning, neural networks and artificial intelligence, ISBN 978-1-4842-2844-9, 2017.

[22] Fausett, L.V., Fundamentals of neural networks: architectures, algorithms and applications, *Pearson Education India*., ISBN 9780133341867, 2006, p. 461.

[23] Bishop, C.M., Neural networks for pattern recognition, *Oxford university press*, ISBN 19 85 38 64 2, 1995.

[24] Moolayil, J., Moolayil, J. and John, S., Learn Keras for deep neural networks, Berkeley, CA, USA: *Apress*., ISBN 978-1-4842-4239-1, 2019.

[25] D. Elbrächter, D. Perekrestenko, P. Grohs and H. Bölcskei, Deep Neural Network Approximation Theory, *in IEEE Transactions on Information Theory*, vol. 67, no. 5, pp. 2581-2623, 2021.

[26] Villegas-Mier, C.G., Rodriguez-Resendiz, J., Álvarez-Alvarado, J.M., Rodriguez-Resendiz, H., Herrera-Navarro, A.M. and Rodríguez-Abreo, O., Artificial neural networks in MPPT algorithms for optimization of photovoltaic power systems: A review., *Micromachines*, vol. 12, No. 10, 2021, p.1260.

[27] Trends, perspectives, and prospects. *Science*, Vol. 349, No. 6245, 2015, pp.255-260.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.