The Journal of Supercomputing (2024) 80:19462–19484
https://doi.org/10.1007/s11227-024-06211-9

A novel hybrid deep learning model for early stage diabetes risk prediction

Mehmet Akif Bülbül[1]

Accepted: 8 May 2024 / Published online: 27 May 2024

Abstract
Diabetes is a prevalent global disease that significantly diminishes the quality of life and can even lead to fatalities due to its complications. Early detection and treatment of diabetes are crucial for mitigating and averting associated risks. This study aims to facilitate the prompt and straightforward diagnosis of individuals at risk of diabetes. To achieve this objective, a dataset for early stage diabetes risk prediction from the University of California Irvine (UCI) database, widely utilized in the literature, was employed. A hybrid deep learning model comprising genetic algorithm, stacked autoencoder, and Softmax classifier was developed for classification on this dataset. The performance of this model, wherein both the model architecture and all hyperparameters were specifically optimized for the given problem, was compared with commonly used methods in the literature. These methods include K-nearest neighbor, decision tree, support vector machine, and convolutional neural network, utilizing tenfold cross-validation. The results obtained with the proposed method surpassed those obtained with other methods, with higher accuracy rates than previous studies utilizing the same dataset. Furthermore, based on the study's findings, a web-based application was developed for early diabetes diagnosis.

Keywords Genetic algorithm · Stacked autoencoder · Softmax classification · Hyperparameter optimization · Hybrid deep learning model

1 Introduction

Diabetes, a chronic metabolic disease, is associated with disturbances in protein, glucose and fat metabolism caused by relative or absolute insulin deficiency [1]. The disease can be diagnosed by high blood glucose levels due to a deficiency in insulin production [2]. There are two different types of diabetes, Type 1 (T1D) and Type 2 (T2D). Diabetes affects an estimated 537 million adults worldwide [3]. If

✉ Mehmet Akif Bülbül
makifbulbul@kayseri.edu.tr

[1] Department of Software Engineering, Kayseri University, 38280 Kayseri, Turkey

this trend continues, it is estimated that the number of diabetics worldwide could exceed 629 million by 2045 [4]. According to the World Health Organization, diabetes is a worldwide epidemic [5]. Approximately 760 billion dollars are spent annually in the fight against diabetes [6]. The cost of treating diabetes-related complications imposes a significant economic burden on healthcare systems, and this burden is increasing day by day [7].

With the widespread prevalence of diabetes, machine learning (ML) models are increasingly used to predict diabetes and its possible complications due to their ability to deal with large and complex datasets. Tan et al. [8] used ML methods to predict T2D diabetes complications. As a result of the study, they obtained successful results in the T2D estimation of the random forest (RF) method. Grupta et al. [9] have proposed a Deep Density Layer Neural Network (DDLNN) for diabetes prediction. Naive Bayes (NB), K-nearest neighbor (KNN), logistic regression (LR), support vector machine (SVM) and decision tree (DT), which are ML algorithms frequently used in the literature, were also used in the study. In line with the findings obtained as a result of the study, it was emphasized that the proposed model gave more effective results than other machine learning methods. Popo and Khosa [10] estimated blood glucose in patients with T1D over a 30- and 60-minute progressive period. They used a multilayer long-short-term memory-based recurrent neural network for blood glucose prediction and obtained successful results. It was emphasized that the findings obtained from this study will shed light on the determination of the amount of insulin to be given to T1D patients. Zheng et al. [11] proposed a multivariate risk prediction model for the prediction of T2D patients. The proposed model is based on 5 independent influence factors and the LR method is used. Successful results were obtained in the study, in which the clinical features of T2D disease were shown comprehensively. Zhu et al. [12] [deep learning and meta-learning] propose a fast-adaptive and confident neural network (FCNN) to predict blood glucose in T1D patients. The proposed method was developed on 3 different data sets. The findings obtained as a result of the study produced successful results in predicting the future periods of blood sugar in the 18- and 64-minute periods. Alqushaibi et al. [13] created a hybrid structure using Bayesian optimization algorithm and convolutional neural network (CNN) for T2D risk prediction. Successful results were obtained in the Bayesian CNN architecture. Li et al. [14] developed a diabetes risk prediction model based on XGBoost. Integrated learning, deep learning, and logistic regression methods were used in the developed method. An AUC value of 0.91 was obtained with the proposed method. Aslan and Sabancı [15] proposed a new method based on deep learning for diabetes prediction. In the study, firstly, numerical data were converted into visuals. The resulting images are first given to CNN's ResNet-18 and ResNet-50 models. Then, the features of the ResNet model are combined with SVM and classified. In the last step, the fusion features are classified according to SVM. Successful results were obtained with the proposed method. Nguyen et al. [7] used KNN, LR, SVC, AdaBoost Classifier, Gradient Enhancer Classifier, and RF Classifier methods to diagnose T2D patients. In the study where the results were presented in a controversial manner, it was emphasized that RF produced more successful results than other ML methods. Naz and

Ahuja [16] proposed an expert system for T2D diabetes detection. The proposed method is a hybrid of synthetic minority oversampling technique (SMOTE) and sequential minimal optimization (SMO) algorithms. SMOTE was used for data preprocessing, and SMO was used for classification. An accuracy rate of 99.07% was achieved with the proposed method.

Although various methods for predicting diabetes are frequently employed in the literature, their predictive performance remains limited due to challenges such as hyperparameter selection and parameter optimization [13, 15, 17]. The choice of hyperparameters plays a crucial role in enhancing the classification performance of these methods [18]. Adjusting problem-specific hyperparameters through trial-and-error methods is nearly impossible [19, 20]. Hybrid structures of the methods used for hyperparameter selection, which constitute a multidimensional space, with optimization algorithms will increase the prediction capabilities of these methods [21]. Hence, the efficacy of structures constructed using hyperparameters determined solely through trial and error on particular models remains constrained. Furthermore, existing literature lacks comprehensive studies necessary for the development of real-time implementations of successful models. Patient decision support system applications that could potentially be developed serve as pivotal tools in safeguarding and directing patients' health, thereby mitigating various adverse effects of diseases in their early stages. In light of these considerations, the primary objective and purpose of this study are:

- To achieve high accuracy in predicting patients' diabetes risks.
- Developing a hybrid architecture using stacked autoencoders and Softmax and then optimizing both the structure and hyperparameters of the architecture using genetic algorithm.
- It is to offer a hybrid methodology designed to optimize entire architectures and hyperparameters tailored to specific problems and datasets.

According to these goals and purposes:

- To enhance classifier efficacy through synergizing the unique strengths of stacked autoencoder, Softmax, and genetic algorithm, we devised a deep learning network rooted in optimization and hybridization with the aforementioned components. This network amalgamates the capabilities of stacked autoencoder, Softmax, and genetic algorithm to yield a more potent classifier.
- In order to get effective problem-specific outcomes, GA was used to optimize both the architecture parameters and the hyperparameters inside the architecture.
- The proposed hybrid deep learning model was applied to the early stage diabetes risk prediction dataset from UCI. In addition, different artificial intelligence methods frequently used in the literature were applied to this data set and the findings were presented comparatively.
- The results obtained with the proposed hybrid deep learning model are presented in comparison with the results of other studies conducted with the same data set in the literature.

Novelties that resulted from experiments:

- We developed a hybrid deep learning network approach featuring a problem-specific model and an innovative hyperparameter optimization technique.
- Our approach achieved higher success rates in predicting early stage diabetes risk compared to both the methods used in this study and those documented in the existing literature for this dataset.
- Furthermore, we developed a web-based application for early stage diabetes risk prediction, which can be integrated into patient decision support systems.

2 Materials and methods

2.1 Genetic algorithm

The optimization strategy has been acknowledged in the literature as a means of achieving the best result that is closest to the solution. It permits the determination of unknown parameter values within specific allowable limitations [22]. Often utilized in the literature, GA is an optimization technique that incorporates organic evolutionary processes [23]. The main goal of GA is to produce the best solutions based on the principle of survival of the fittest [24]. The steps of GA are shown in Fig. 1.

In Fig. 1, many random solutions are generated for the problem in the genetic space. Each of these solutions is called an individual. The quality of these individuals is measured by the fitness function [25]. The higher the quality of the individual, the more likely it is to survive both the selection and the next generation. The crossover process

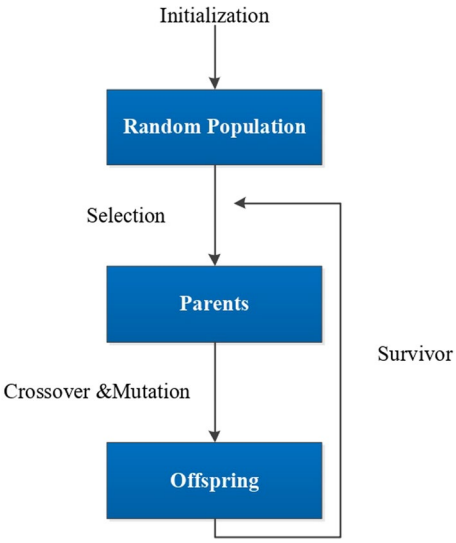


Fig. 1 GA structure

produces new individuals from two selected individuals. Individuals are mutated to maintain diversity in the population [26].

2.2 Stacked autoencoder

SAE is a stacked unsupervised neural network formed by a collection of autoencoder (AE) neural networks. The AE neural network basically consists of two parts: an encoder and a decoder. It converts the input data into features using the encoder and then reconstructs the input data by converting these features back to the initial data through the decoder [27]. A simple AE structure is presented in Fig. 2.

The structure of AE consists of 3 layers as shown in Fig. 2. In AE, the output of each hidden layer is transferred to the next hidden layer as input data. AEs are divided into 2 parts: encoder part and decoder part [28]. AEs reduce the size of the data by feature extraction in the encoder part. Equation (1) is used for this process.

$$y = s\left(W^t x + b\right) \tag{1}$$

In the equation, s represents the Gaussian, sigmoid, and tanh activation functions. W represents the weights between the input layer and the hidden layer. b represents the bias value. x represents the input value. The y is a neuron's scalar output. In the decoder section, the dimensionally reduced data in the hidden layer are decoded. At this stage, the data dimensions are made close to the dimensions of the input data. Equation (2) is used for this process.

$$z = s\left(Wy + b'\right) \tag{2}$$

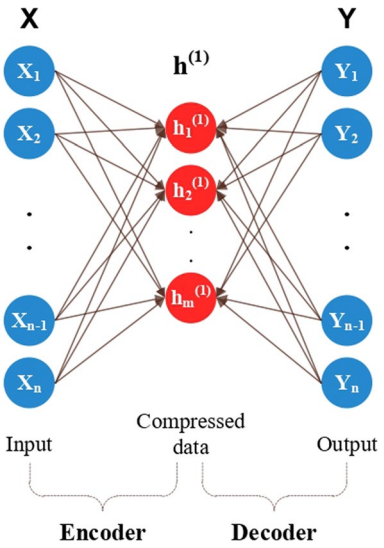


Fig. 2 AE structure

where z represents the reconstructed state of the input values. After this step, a back-propagation algorithm is used to bring the new values close to the input layer data. Equation (3) is used for this.

$$min \sum_{i=1}^{m} (z - x)^2 \tag{3}$$

This process is performed to reveal important data in the data [29].

The overlearning problem within the AE architecture is solved by using the regularization method presented in Eq. (4) [30].

$$min \left[\left(\sum_{j=1}^{m} (x' - x)^2 \right) + \gamma L(w) \right] \tag{4}$$

In Eq. (4), γ represents the regularization parameter. $L(w)$ represents the weight adjustment parameter. Here, the error term is multiplied by the back-propagation algorithm of the weighting factor to avoid overlearning. At this stage, the values of the parameters after sigma are determined by trial and error to obtain the best result.

For more efficient classification, multiple autocoders can be connected to each other. The connected autocoders form a stacked AE. A simple stacked AE is shown in Fig. 3.

As shown in Fig. 3, the output data obtained with stacked AE are used as input data for the Softmax classifier. With this structure, the data in the input layer are classified by the Softmax classifier in the output layer. Softmax classifier is a probability-based linear classifier used when there are two or more classes [30].

The stacked autocoder architecture shown in Fig. 3;

- Number of encoders and decoders
- Number of layers in encoders and decoders
- Activation function used in layers in the encoder section

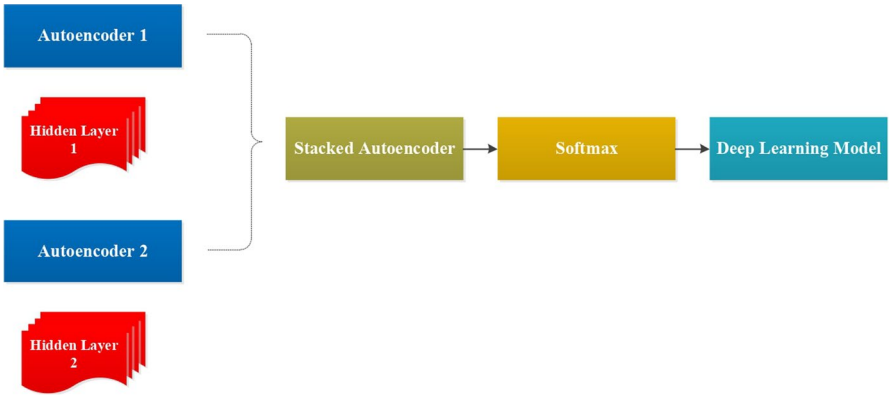


Fig. 3 Stacked autoencoder and deep learning structure

- Activation function used in layers in the decoder section
- Weight regularization coefficient (L2WeightRegularization) value used to prevent overfitting and improve the generalization ability of models
- Sparsity ratio coefficient (SparsityRegularization) value, which helps back-propagation and is used in dilution
- Sparsity adjustment coefficient (SparsityProportion) value, which helps backscatter and is used in dilution
- Lscaledata values directly affect the performance of the architecture.

In order to achieve high success in solving classification problems, it is almost impossible to set these parameters appropriately by trial-and-error methods. It is imperative to establish hybrid structures with optimization algorithms for these parameters, which should be determined specifically for the problem, to achieve high successful results [18].

2.3 Other methods used in experimental studies

The artificial intelligence methods utilized in experimental studies for early stage diabetes risk prediction, commonly found in the literature, are presented in Table 1.

These methods presented in Table 1 are frequently used in classification problems [35–38].

Table 1 Other methods used in experimental studies

Methods	Descriptions
K-Nearest neighbor (KNN)	The KNN algorithm stands as one of the simplest machine learning algorithms. One of its significant advantages lies in its independence from assumptions regarding the distribution of underlying data. This method operates by considering the degree of similarity among nearest neighbors [31]
Decision trees (DTs)	DT is a learning algorithm frequently used in the literature for classification problems. The decision tree consists of branches, leaves and decision nodes [32]
Support vector machine (SVM)	SVM is a supervised algorithm utilized for classification or regression tasks. Primarily employed in classification problems, SVM operates based on the concept of a hyperplane that effectively separates different data classes [33]
Convolutional neural network (CNN)	CNNs, extensively employed in various computer vision domains, particularly in classification tasks, differ from classical neural networks by incorporating convolutional feature extraction and classification layers [34]

3 Experimental studies and results

3.1 Dataset

In the study, the early stage diabetes risk prediction dataset available at the University of California Irvine (UCI) was used for diabetes risk prediction. This dataset is frequently used in the literature [39–41]. The attributes in this dataset are shown in Table 2.

In the dataset, whose attributes are shown in Table 2, there are 16 disease-defining attributes. With these attributes on the dataset, it is shown whether people belong to the diabetes class or not. There are a total of 520 patients in the dataset, 192 women and 328 men. While 200 of these patients were healthy, 320 of them were diagnosed with diabetes.

The data in the dataset used in the study are divided into two groups as 70% training and 30% testing.

3.2 Performance evaluations matrices

In this study, accuracy, sensitivity, precision, accuracy, and F1 score values will be used to determine the success of each classifier. In order to calculate these values, the complexity matrix of the classifiers must first be created. For a 2-class classification, the complexity matrix is shown in Fig. 4.

In Fig. 4, TP represents true positives, that is, the number of healthy individuals estimated to be healthy. FP represents true negatives, that is, the number of sick

Table 2 Dataset and attributes

Attribute	Data Type	Values
Age	Input	20–65
Sex	Input	Female /male
Polyuria	Input	No/yes
Polydipsia	Input	No/yes
Sudden weight loss	Input	No/yes
Weakness	Input	No/yes
Polyphagia	Input	No/yes
Genital thrush	Input	No/yes
Visual blurring	Input	No/yes
Itching	Input	No/yes
Irritability	Input	No/yes
Delayed healing	Input	No/yes
Partial paresis	Input	No/yes
Muscle stiffness	Input	No/yes
Alopecia	Input	No/yes
Obesity	Input	No/yes
Class	Output	Negative/positive

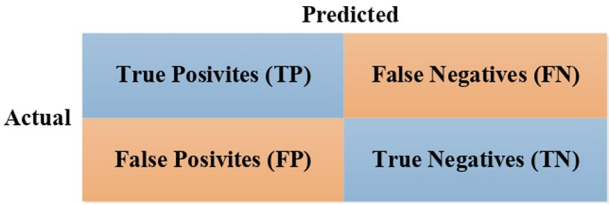


Fig. 4 2-class complexity matrix

individuals estimated to be healthy. TN represents false positives, i.e., the number of sick individuals predicted to be sick. FN represents false negatives, that is, the number of healthy individuals predicted to be sick.

The calculation of accuracy, precision, F1 score, and recall based on a complexity matrix shown in Fig. 4 is shown in Eqs. (5–8).

$$Accuracy\,(A) = \frac{TP + TN}{TP + TN + FN + FP} \tag{5}$$

$$Precision\,(P) = \frac{TP}{TP + FP} \tag{6}$$

$$Recall\,(R) = \frac{TP}{TP + FN} \tag{7}$$

$$F1score = \frac{2 * P * R}{P + R} \tag{8}$$

3.3 Proposed deep learning model

In this section, a hybrid deep learning model architecture was created with GA, SAE and Softmax classifier. The flow diagram of the architecture is shown in Fig. 5.

In Fig. 5, initially, SAE architectures will be generated with various configurations and hyperparameters within defined limits corresponding to the number of populations. Each architecture will be paired with a Softmax classifier. Subsequently, these composite structures will undergo training followed by testing on previously unseen data. Throughout each iteration of the GA, both the architectural parameters and hyperparameters within the architectures will be optimized to enhance prediction accuracy, leveraging the capabilities of the GA.

Most of the information in the data set used in the flow diagram presented in Fig. 5 has first been preprocessed. In this preprocessing step, 364 data points were randomly selected from the dataset for the training dataset. The remaining 156 data points were allocated as the test dataset for evaluating each gene within the architecture. The representations of the data used in the proposed model are shown in Table 3.

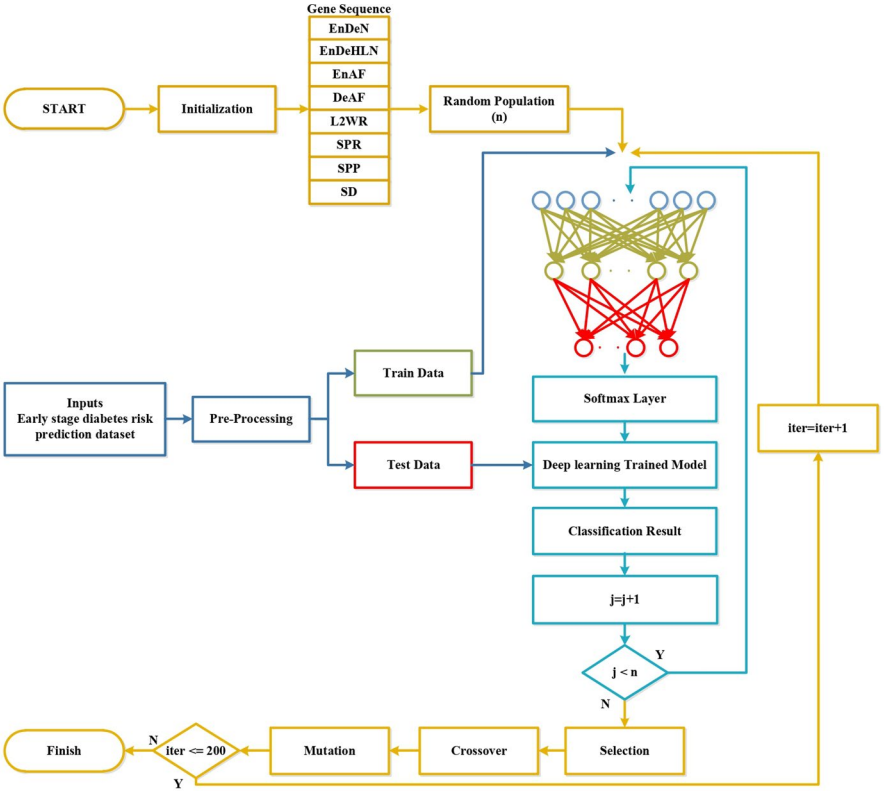


Fig. 5 Hybrid deep learning model

The dataset comprises 16 attributes that evaluate the risk of early stage diabetes, resulting in a classification area derived from these features. The presented model aims to make predictions utilizing these 16 attributes. The representation of the attributes in the dataset shown in Table 3 will provide ease of coding.

In the flow diagram presented in Fig. 5, the initial parameters are determined as the first step. In the proposed hybrid deep learning model, the initial parameters were determined as a result of experimental studies and these parameters are shown in Table 4. The proposed hybrid deep learning model was developed in MATLAB platform due to the ease of coding with multilayer data.

The parameters presented in Table 4 are necessary for the GA to function in the proposed hybrid deep learning model. After determining the initial parameters, the next step for the proposed model is to create the initial population. In the proposed hybrid deep learning model, each gene contains parameters that can create SAE and Softmax classifiers. The chromosome sequence of each gene is shown in Table 5.

These parameters, presented in Table 5 and present in each gene structure, are determined randomly. The *EnDeN* value for each gene (G) is randomized according to the constraint function presented in Eq. (9).

Table 3 Data preprocessing

Attribute	Values	Representation in the proposed model
Age	Integer	Integer
Gender	Female /male	0/1
Polyuria	No/yes	0/1
Polydipsia	No/yes	0/1
Sudden weight loss	No/yes	0/1
Weakness	No/yes	0/1
Polyphagia	No/yes	0/1
Genital thrush	No/yes	0/1
Visual blurring	No/yes	0/1
Itching	No/yes	0/1
Irritability	No/yes	0/1
Delayed healing	No/yes	0/1
Partial paresis	No/yes	0/1
Muscle stiffness	No/yes	0/1
Alopecia	No/yes	0/1
Obesity	No/yes	0/1
Class	Negative/positive	0/1

Table 4 GA parameters for hybrid deep learning model

Algorithm parameters	Values
Population number	40
Solution space	8
Selection rate	0.85
Mutation rate	0.03
Iteration number	200

Table 5 Chromosome Sequence

Parameters	Values
EnDeN	Number of encoders and decoders
EnDeHLN	Encoder and decoder hidden layer numbers
EnAF	Activation function used in encoder layers
DeAF	Activation function used in decoder layers
L2WR	L2 weight regularization parameter
SPR	Sparsity regularization parameter
SPP	Sparsity proportion parameter
SD	Lscaledata value

$$G_i\,En\,De\,N_j(x) = \begin{cases} 1 & x < 1 \\ x & 1 \le x \le 3 \\ 3 & x > 3 \end{cases} \tag{9}$$

The $G_iEnDeN_j(x)$ given in Eq. (9) represents the *EnDeN* value of gene i in the j iteration.

The value of *EnDeHLN* for each G is randomly determined according to the constraint function presented in Eq. (10).

$$G_i\,En\,De\,HLN_j(x) = \begin{cases} 5 & x < 5 \\ x & 5 \le x \le 100 \\ 100 & 100 < x \end{cases} \tag{10}$$

The $G_iEnDeHLN_j(x)$ given in Eq. (10) represents the *EnDeHLN* value of gene i in the j iteration.

The *EnAF* value for each G is randomly determined according to the constraint function presented in Eq. (11).

$$G_i\,En\,AF_j(x) = \begin{cases} 1 & x < 1 \\ x & 1 \le x \le 2 \\ 2 & 2 < x \end{cases} \tag{11}$$

The $G_iEnAF_j(x)$ given in Eq. (11) represents the *EnAF* value of gene i in the j iteration. The activation functions corresponding to the determined $G_iEnAF_j(x)$ value are shown in Table 6.

The *DeAF* value for each G is randomly determined according to the constraint function presented in Eq. (12).

$$G_i\,De\,AF_j(x) = \begin{cases} 1 & x < 1 \\ x & 1 \le x \le 3 \\ 3 & x > 3 \end{cases} \tag{12}$$

The $G_iDeAF_j(x)$ given in Eq. (12) represents the *DeAF* value of gene i in the j iteration. The activation functions corresponding to the determined $G_iDeAF_j(x)$ value are shown in Table 7.

Table 6 EnAf parameters and values

EnAf values	Corresponding activation function
1	Logsig
2	Satlin

Table 7 DeAf parameters and values

DeAf values	Corresponding activation function
1	Logsig
2	Satlin
3	Purelin

The *L2WR* value for each G is randomly determined according to the constraint function presented in Eq. (13).

$$G_i\, L2\, WR_j(x) = \begin{cases} 0.001 & x < 0.001 \\ x & 0.001 \le x \le 0.01 \\ 0.01 & x > 0.01 \end{cases} \tag{13}$$

The $G_iL2WR_j(x)$ given in Eq. (13) represents the *L2WR* value of gene i in the j iteration.

The *SPR* value for each G is randomized according to the constraint function presented in Eq. (14).

$$G_i\, SPR_j(x) = \begin{cases} 1 & x < 1 \\ x & 1 \le x \le 5 \\ 5 & x > 5 \end{cases} \tag{14}$$

The $G_iSPR_j(x)$ given in Eq. (14) represents the *SPR* value of gene i in the j iteration.

The *SPP* value for each G is randomly determined according to the constraint function presented in Eq. (15).

$$G_i\, SPP_j(x) = \begin{cases} 0 & x < 0 \\ x & 0 \le x \le 1 \\ 1 & x > 1 \end{cases} \tag{15}$$

The $G_iSPP_j(x)$ given in Eq. (15) represents the *SPP* value of gene i in the j iteration.

The *SD* value for each G is randomly determined according to the constraint function presented in Eq. (16).

$$G_i\, SD_j(x) = \begin{cases} 1 & x < 1 \\ x & 1 \le x \le 2 \\ 2 & x > 2 \end{cases} \tag{16}$$

The $G_iSD_j(x)$ given in Eq. (16) represents the *SD* value of gene i in the j iteration. The corresponding value in the hybrid structure for the determined $G_iSD_j(x)$ value is shown in Table 8.

Table 8 SD parameters and values

SD values	Corresponding values
1	True
2	False

In the next step of the proposed hybrid deep learning model, each G performs a deep network learning according to the values it contains. The training section reserved on the data set is used for learning. After the SAE learning phase created by each G, they were put into the testing phase with the remaining 30% of the data set. The test results reveal the performance of the constructs created for each G. The SAE or AE performance of each G is determined according to the fitness functions given in Eqs. (17–19).

$$Traf(G_i) = Learning - Accuracy\left(YAENCO_i\right) \tag{17}$$

$$Valf(G_i) = Test - Accuracy\left(YAENCO_i\right) \tag{18}$$

$$Fitf(G_i) = Max\left(Traf(G_i) + Valf\left(YAENCO_i\right)\right) \tag{19}$$

$YAENCO_i$ given in the equations, i. represents the SAE or AE generated by the gene. $Traf(G_i)$ represents the relevance of the SAE or AE generated by gene i in learning. $Valf(G_i)$ represents the suitability of the SAE or AE generated by the i gene at the test stage. $Fitf(G_i)$ represents the fitness value of i gene.

The next step of the proposed hybrid deep learning model is the selection process. In the selection process, the cumulative *Fitf* values of all G's in the population are calculated and placed on the roulette wheel according to these values. In the selection process, random roulette values were determined and as many genes as the number of populations shown in Table 4 were selected. In this step, the roulette wheel method was chosen because it increases the survival probability of genes with high fitness values and produces successful results.

The next step in the proposed hybrid deep learning model is the crossover process. In this step, random genes were randomly selected in pairs for the crossover process and the crossover process was performed among them. The crossover process between genes was performed over a single randomly determined point.

The next step in the proposed hybrid deep learning model is the mutation process. In this step, a random mutation value is generated for each chromosome in each gene. This value generated for each chromosome in each gene was compared with the mutation value shown in Table 4, and the chromosomes to be mutated were determined. While applying the mutation process to the chromosomes in the genes, the restriction functions presented in Eqs. (9–16) were also taken into account.

As a result of the processes executed in each iteration within the hybrid deep learning model, each gene possesses a distinct architecture, varying numbers of autoencoders, and diverse hyperparameters. Consequently, during each iteration,

Table 9 The most successful gene chromosomes

Kromozom parameters	Values
EnDeN	1
EnDeHLN	46
EnAF	1 (logsig)
DeAF	3 (Purelin)
L2WR	0,001
SPR	3.061129140957694
SPP	0.496217506819641
SD	2 (false)

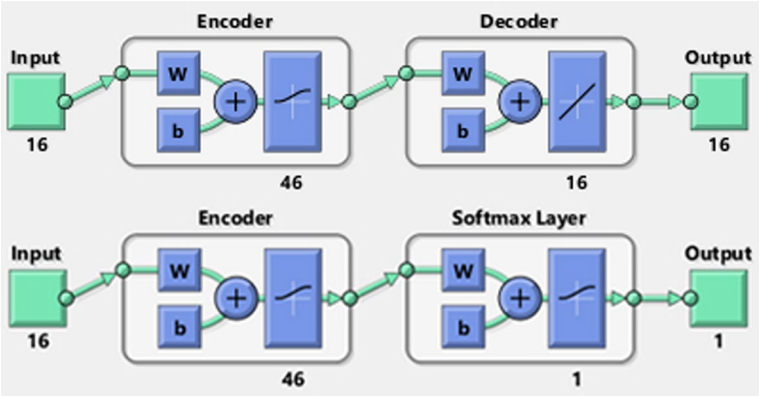


Fig. 6 An AE and Softmax classifier structure

models with disparate architectures and hyperparameters are trained using the training dataset and evaluated with the test dataset to ascertain their effectiveness.

The proposed hybrid deep learning model was run for 200 iteration as shown in Table 4. After the run, the most successful gene in the population was determined according to the fitness functions given in Eqs. (17–19). The chromosome values in the most successful gene according to the fitness values are shown in Table 9.

An AE and Softmax classifier structure built with the parameters presented in Table 9 is shown in Fig. 6.

4 The experimental results and discussion

In this study for early stage diabetes risk prediction, KNN, DT, SVM, and CNN clustering methods were used along with the hybrid deep learning model presented in this study. The complexity matrix of each model used in the study is shown in Fig. 7.

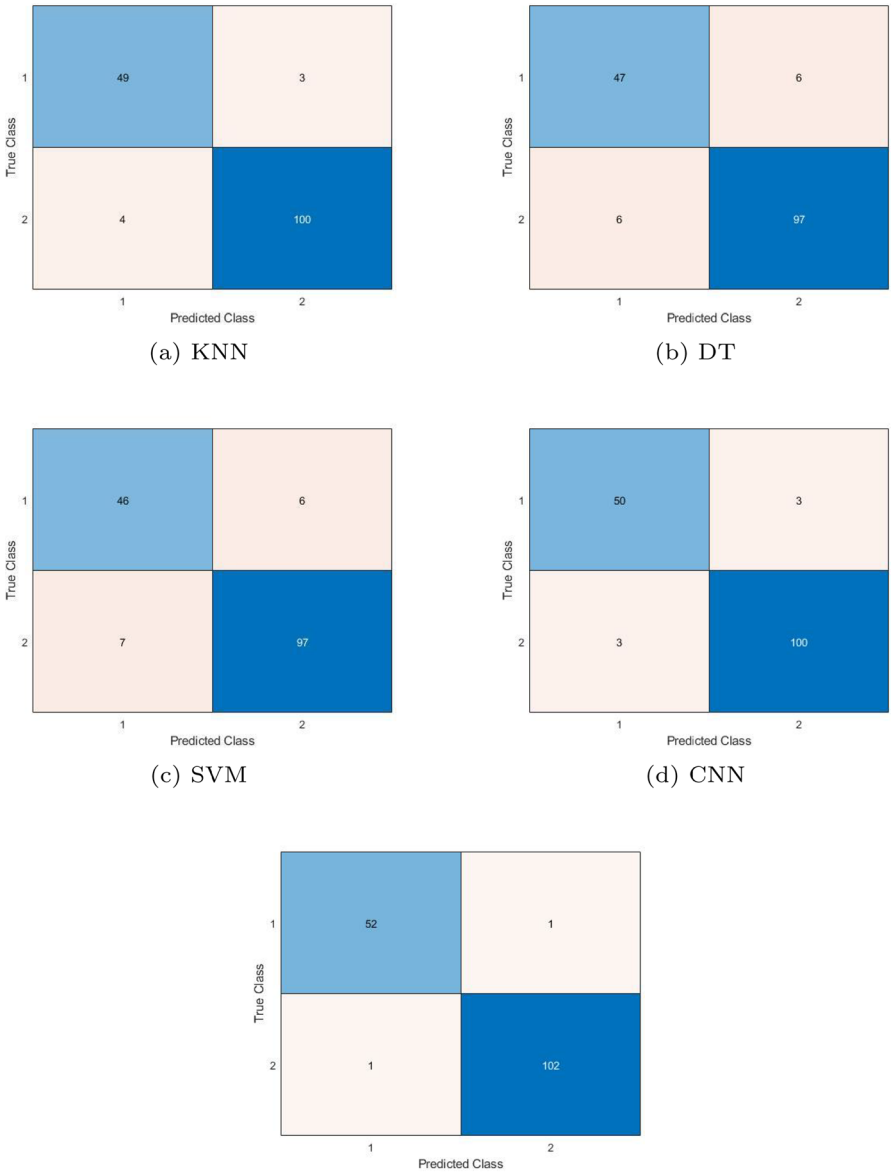
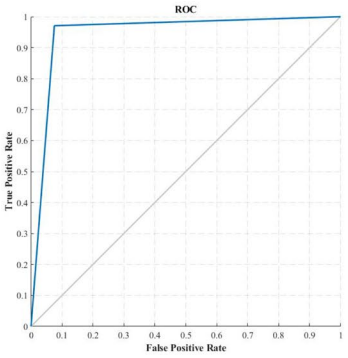
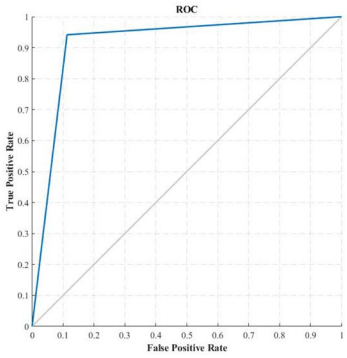


Fig. 7 Confusion matrices of the models used

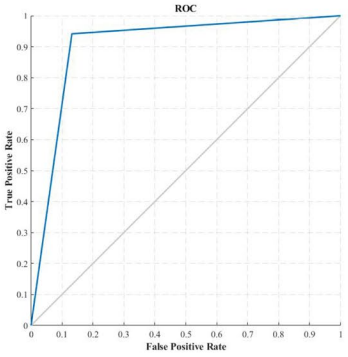
In the confusion matrices depicted in Fig. 7, the SVM model exhibited the poorest performance in classifying early stage diabetes risk. Conversely, the hybrid deep learning model emerged as the most accurate predictor for diagnosing the disease. According to the values presented in Fig. 7, the proposed hybrid deep learning model performed a more successful classification process than the other classifiers.



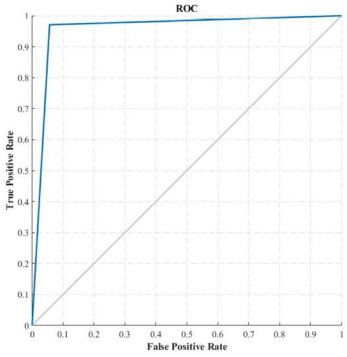
(a) KNN



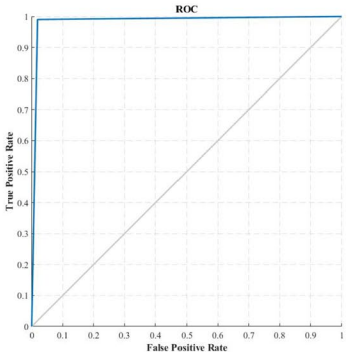
(b) DT



(c) SVM



(d) Hybrid deep learning model



(e) Hybrid deep learning model

Fig. 8 ROC curves of the models used

Table 10 Results for diabetes classification

Model	Precision (%)	Recall (%)	F1 Score (%)	Accuracy (%)
KNN	94.23	92.45	93.33	95.51
DT	88.67	88.67	88.67	92.30
SVM	88.46	88.79	87.61	91.66
CNN	94.34	94.34	94.34	96.15
Hybrid deep learning model	98.11	98.11	98.11	98.72

Table 11 Literature comparison with the proposed hybrid deep learning model

Study	Method	Accuracy (%)
Ferdousi et al. [39]	Random Tree	94
Wijayaningrum et al. [42]	MLP	95.38
Tan et al. [43]	RF	97
Laila at al. [41]	RF	97.11
Yılmaz [40]	GA-XGBoost	97.23
Proposed model	GA-SAE-Softmax	98.72

The ROC curves of each classifier according to the complexity matrices shown in Fig. 7 are presented in Fig. 8.

The ROC curve shows the false-positive rate on the x-axis and the true-positive rate on the y-axis. The AUC (area under the curve) value is the area under the ROC curve and is used to measure the success of a model. The larger the AUC, the better the performance of the model. The best AUC value is 1. Figure 8 shows that the presented hybrid deep learning model performs the best compared to other methods.

Table 10 shows the results of the methods used to classify the early stage diabetes risk prediction dataset according to the evaluation metrics given in Eqs. (17–19).

In Table 10, the SVM model yielded the lowest accuracy rate, followed by DT, KNN, and then CNN models. Conversely, the proposed hybrid deep learning model achieved the highest accuracy. When the values given in Table 10 are examined, it is seen that the proposed hybrid deep learning model produces more successful results than other methods. The results obtained with the hybrid deep learning model proposed with the previous studies in the literature on the early stage diabetes risk prediction dataset are shown in Table 11.

As can be seen in Table 11, the proposed hybrid deep learning model has produced more successful results than the previous studies in the literature.

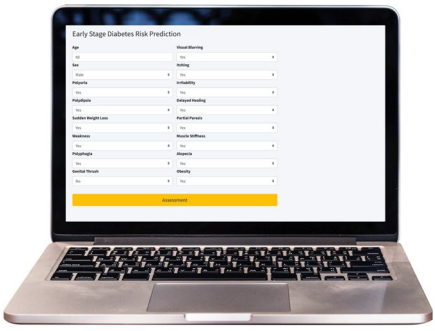
5 Application

In this part of the study, a web-based application for early stage diabetes risk prediction is developed using the findings from the experimental studies. First of all, a functional architecture was coded in MATLAB platform using the parameters

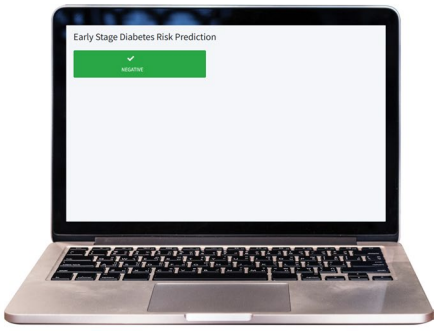
obtained with the proposed hybrid deep learning model. The structure created as a function in MATLAB platform takes the data presented in Table 2 as parameters, respectively. The function sends back the result 0 or 1 according to the data it receives as parameters. The prepared function was converted into a DLL file on the MATLAB platform. This DLL file was used in the web-based software developed in Visual Studio 2022 Community platform. While developing the web-based application, C-Sharp programming language was chosen as the coding language and Visual Studio 2022 Community was chosen as the platform due to its free and easy interface. With the DLL file added to the Reference section of the created application project, an early stage diabetes risk prediction application developed with C-Sharp programming language was created. The interfaces of the created application are shown in Fig. 9.

In Fig. 9a, the parameters used as input in the proposed hybrid deep learning model are received from the user. These parameters received from the user are sent to the function in the DLL through the application. The user is informed in Fig. 9b and c according to the 0 or 1 value returned as a result of the processing of these parameters sent to the generated DLL file.

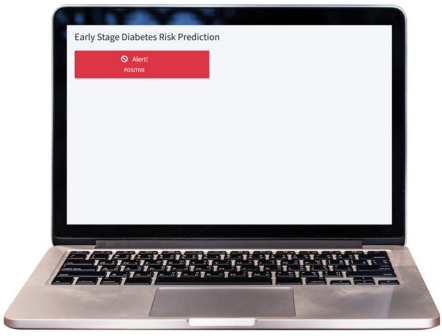
The early stage diabetes risk prediction application showcased in Fig. 9 is web-based and designed for seamless integration into medical field applications. Consequently, it holds substantial potential for widespread adoption and utilization.



(a) Information entry



(b) Information interface negative



(c) Information interface positive

Fig. 9 Web-based application interfaces

6 Conclusion and discussion

Diabetes is a disease that has turned into a worldwide epidemic, affects the quality of life of people and other body organs if no precautions are taken, and can even result in death in the very advanced stages of the disease. Measures to be taken with the early diagnosis of this disease can eliminate the bad consequences of this disease. In this article, a hybrid deep learning model based on GA-SAE-Softmax classifier is proposed for early stage diabetes risk prediction. Early stage diabetes risk estimation data set, which is frequently used in the literature and taken from UCI, was used as the data set in the proposed hybrid deep learning model. In the proposed deep learning model, SAE and Softmax classifier are combined and hyperparameters in the structure are optimized with GA in order to maximize the performance of the created architecture. In addition, different methods were applied to the same data set. Following the experimental studies, the proposed hybrid deep learning model achieved a prediction accuracy of 98.72% on the dataset. Experimental results have shown that the proposed deep learning model gives better results than other methods, as well as better results than studies conducted on the same data set so far.

With the proposed hybrid deep learning model;

- A model with higher accuracy than the studies in the literature has been created by using the attributes in the early stage diabetes risk prediction dataset.
- The number of encoders and decoders that should be used in an architecture built with SAE and Softmax classifier in early stage diabetes risk prediction, the number of hidden layers in the encoder and decoder, the activation functions to be used in the layers in the encoder section, the activation functions to be used in the layers in the decoder section, the weight adjustment coefficient used to prevent overfitting of the model and to increase the generalization sail, the sparsity ratio coefficient that helps back-propagation in the model and is used in dilution, the sparsity adjustment coefficient were determined by optimizing as a whole.
- In addition to optimizing all the parameters required for a new architecture to be created, the proposed model also provides access to these parameter values.
- The proposed hybrid deep learning model distinguishes itself from other methods by crafting problem-specific structures capable of achieving high success rates through the fusion of prominent features from various methods incorporated within the model.

With the web-based application prepared by using the values of the parameters determined by the hybrid deep learning model as a result of experimental studies;

- Doctors who are not experts in the field will be able to perform diabetes risk prediction.
- With the application, early diagnosis will be made and necessary precautions will be taken before the disease progresses, and diabetes that will cause serious consequences will be prevented.

- Thanks to the early diagnosis application, it will significantly save the use of medication that patients need to use in the later stages, the cost and time to be spent for these drugs.
- It will prevent serious health problems that will occur with the progression of diabetes and the death rate caused by the disease will decrease.

This developed web-based application can be applied as an alternative method in health decision support systems.

The limitation of this study is bound by the size of the dataset utilized. To enhance the precision and effectiveness of predictions and diagnoses, it is advisable to generate larger datasets and subsequently refine them for further analysis.

In future studies, an application can be developed for smartphones without the need for internet in the light of the findings obtained from this study.

Author contributions The authors contributed equally to the work.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Declarations

Conflict of interest The authors declare that there is no conflict of interest between them.

Ethical approval The authors declare that no ethical approval is required for this study.

References

1. Nnamudi AC, Orhue NJ, Ijeh II, Nwabueze AN (2023) Finnish diabetes risk score outperformed triglyceride-glucose index in diabetes risk prediction. J Diabetes Metab Disord 22(2):1337–1345
2. Cheng H, Zhu J, Li P, Xu H (2023) Combining knowledge extension with convolution neural network for diabetes prediction. Eng Appl Artif Intell 125:106658
3. Kumar A, Gangwar R, Ahmad Zargar A, Kumar R, Sharma A (2024) Prevalence of diabetes in India: a review of IDF diabetes atlas 10th edition. Curr Diabetes Rev 20(1):105–114
4. Yuan Z, Ding H, Chao G, Song M, Wang L, Ding W, Chu D (2023) A diabetes prediction system based on incomplete fused data sources. Mach Learn Knowl Extr 5(2):384–399
5. Organization WH et al. (2004) Diabetes action now: an initiative of the world health organization and the international diabetes federation
6. Zhou H, Xin Y, Li S (2023) A diabetes prediction model based on Boruta feature selection and ensemble learning. BMC Bioinf 24(1):1–34

7. Nguyen LP, Tung DD, Nguyen DT, Le HN, Tran TQ, Binh TV, Pham DTN (2023) The utilization of machine learning algorithms for assisting physicians in the diagnosis of diabetes. Diagnostics 13(12):2087
8. Tan KR, Seng JJB, Kwan YH, Chen YJ, Zainudin SB, Loh DHF, Liu N, Low LL (2023) Evaluation of machine learning methods developed for prediction of diabetes complications: a systematic review. J Diabetes Sci Technol 17(2):474–489
9. Gupta N, Kaushik B, Imam Rahmani MK, Lashari SA (2023) Performance evaluation of deep dense layer neural network for diabetes prediction. Comput Mater Contin 76(1)
10. Butt H, Khosa I, Iftikhar MA (2023) Feature transformation for efficient blood glucose prediction in type 1 diabetes mellitus patients. Diagnostics 13(3):340
11. Zheng J, Shen S, Xu H, Zhao Y, Hu Y, Xing Y, Song Y, Wu X (2023) Development and validation of a multivariable risk prediction model for identifying ketosis-prone type 2 diabetes. J Diabetes 15(9):753–764
12. Zhu T, Li K, Herrero P, Georgiou P (2022) Personalized blood glucose prediction for type 1 diabetes using evidential deep learning and meta-learning. IEEE Trans Biomed Eng 70(1):193–204
13. Alqushaibi A, Hasan MH, Abdulkadir SJ, Muneer A, Gamal M, Al-Tashi Q, Taib SM, Alhussian, H (2023) Type 2 diabetes risk prediction using deep convolutional neural network based-bayesian optimization. Comput Mater Contin 75(2)
14. Li L, Cheng Y, Ji W, Liu M, Hu Z, Yang Y, Wang Y, Zhou Y (2023) Machine learning for predicting diabetes risk in western China adults. Diabetol Metab Syndr 15(1):1–12
15. Aslan MF, Sabanci K (2023) A novel proposal for deep learning-based diabetes prediction: converting clinical data to image data. Diagnostics 13(4):796
16. Naz H, Ahuja S (2022) SMOTE-SMO-based expert system for type II diabetes detection using PIMA dataset. Int J Diabetes Dev Ctries 42(2):245–253
17. Bülbül MA (2024) Optimization of artificial neural network structure and hyperparameters in hybrid model by genetic algorithm: iOS-android application for breast cancer diagnosis/prediction. J Supercomput 80(4):4533–4553
18. Bülbül MA, Öztürk C (2022) Optimization, modeling and implementation of plant water consumption control using genetic algorithm and artificial neural network in a hybrid structure. Arab J Sci Eng 47(2):2329–2343
19. Işık MF, Avcil F, Harirchian E, Bülbül MA, Hadzima-Nyarko M, Işık E, İzol R, Radu D (2023) A hybrid artificial neural network-particle swarm optimization algorithm model for the determination of target displacements in mid-rise regular reinforced-concrete buildings. Sustainability 15(12):9715
20. Khetavath S, Sendhilkumar NC, Mukunthan P, Jana S, Gopalakrishnan S, Malliga L, Chand SR, Farhaoui Y (2023) An intelligent heuristic manta-ray foraging optimization and adaptive extreme learning machine for hand gesture image recognition. Big Data Mining Anal 6(3):321–335
21. Bülbül MA (2023) A hybrid approach for multiclass classification of dry bean seeds. J Inst Sci Technol 13(1):33–43
22. Konak F, Bülbül MA, Türkoğlu D (2024) Feature selection and hyperparameters optimization employing a hybrid model based on genetic algorithm and artificial neural network: Forecasting dividend payout ratio. Comput Econ 1–21
23. Rabee F, Hussain ZM (2023) Oriented crossover in genetic algorithms for computer networks optimization. Information 14(5):276
24. Salto C, Minetti G, Alba E, Luque G (2023) Big optimization with genetic algorithms: Hadoop, Spark, and MPI. Soft Comput 27(16):11469–11484
25. Wang H, Xu S, Hu H (2023) PID controller for PMSM speed control based on improved quantum genetic algorithm optimization. IEEE Access
26. Chen S (2023) Design of computer big data processing system based on genetic algorithm. Soft Comput 27(11):7667–7678
27. Wang C, Tang X, Yu J, Yang X, Yan X (2024) Mechanistic block-based attention mechanism stacked autoencoder for describing typical unit connection industrial processes and their monitoring. Can J Chem Eng 102(1):291–306
28. Zhang C, Zhang Y, Huang Q, Zhou Y (2023) Intelligent fault prognosis method based on stacked autoencoder and continuous deep belief network. In: Actuators, MDPI. 12:117
29. Baştürk A, Yüksei ME, Badem H, Çalışkan A (2017) Deep neural network based diagnosis system for melanoma skin cancer. In: 2017 25th Signal Processing and Communications Applications Conference (SIU), IEEE. 1–4

30. Adem K, Kiliçarslan S, Cömert O (2019) Classification and diagnosis of cervical cancer with stacked autoencoder and softmax classification. Expert Syst Appl 115:557–564
31. Zhang J, Li Y, Shen F, He Y, Tan H, He Y (2024) Hierarchical text classification with multi-label contrastive learning and KNN. Neurocomputing 577:127323
32. Briglia G, Immovilli F, Cocconcelli M, Lippi M (2023) Bearing fault detection and recognition from supply currents with decision trees. IEEE Access
33. Panigrahi BS, Nagarajan N, Prasad KDV, Salunkhe SS, Kumar P, Kumar MA, et al (2024) Novel nature-inspired optimization approach-based SVM for identifying the android malicious data. Multimed Tools Appl, 1–19
34. Cai J, Boust C, Mansouri A (2024) ATSFCNN: a novel attention-based triple-stream fused CNN model for hyperspectral image classification. Mach Learn Sci Technol. https://doi.org/10.1088/2632-2153/ad1d05
35. Sirhan M, Bekhor S, Sidess A (2024) Multilabel CNN model for asphalt distress classification. J Comput Civ Eng 38(1):04023040
36. Tian G, Wang J, Wang R, Zhao G, He C (2024) A multi-label social short text classification method based on contrastive learning and improved ml-KNN. Expert Syst. https://doi.org/10.1111/exsy.13547
37. Tabany M, Gueffal M (2024) Sentiment analysis and fake amazon reviews classification using SVM supervised machine learning model. J Adv Inf Technol
38. Adeniyi AE, Ayoola JB, Farhaoui Y, Awotunde JB, Imoize AL, Jimoh GR, Chollom DF (2023) Comparative study for predicting melanoma skin cancer using linear discriminant analysis (LDA) and classification algorithms. In: The International Conference on Artificial Intelligence and Smart Environment, Springer. 326–338
39. Ferdousi R, Hossain MA, El Saddik A (2021) Early-stage risk prediction of non-communicable disease using machine learning in health CPS. IEEE Access 9:96823–96837
40. Yilmaz A (2022) Prediction of type 2 diabetes mellitus using feature selection-based machine learning algorithms. Health Probl Civiliz 16(2):128–139
41. Laila UE, Mahboob K, Khan AW, Khan F, Taekeun W (2022) An ensemble approach to predict early-stage diabetes risk using machine learning: an empirical study. Sensors 22(14):5247
42. Wijayaningrum V, Saragih T, Putriwijaya N (2021) Optimal multi-layer perceptron parameters for early stage diabetes risk prediction. In: IOP Conference Series: Materials Science and Engineering, IOP Publishing. vol. 1073, p 012070
43. Tan Y, Chen H, Zhang J, Tang R, Liu P (2022) Early risk prediction of diabetes based on GA-stacking. Appl Sci 12(2):632