Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 198 (2022) 388–406
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original articles

A novel hybrid machine learning framework for the prediction of diabetes with context-customized regularization and prediction procedures

Aghila Rajagopal[a], Sudan Jha[b], Ramachandran Alagarsamy[c], Shio Gai Quek[d], Ganeshsree Selvachandran[d,*]

[a] *Department of CSBS, Sethu Institute of Technology, Virudhunagar, Tamilnadu, India*
[b] *Department of Computer Science and Engineering, Chandigarh University, Punjab, India*
[c] *Department of CSE, University College of Engineering, Panruti, Tamilnadu, India*
[d] *Faculty of Business and Management, UCSI University, Jalan Menara Gading, 56000 Cheras, Kuala Lumpur, Malaysia*

Received 11 June 2021; received in revised form 21 February 2022; accepted 3 March 2022
Available online 14 March 2022

Abstract

This paper proposes a customized hybrid model of artificial neural network (ANN) and genetic algorithms for an efficient diabetes disease prediction framework. Our customized hybrid model uses an improvised technique of detecting the more visible patterns of relations between the variables. Initially, the input medical dataset is preprocessed using a novel normalization technique that works consistently for all degrees of skewness of data. Then, our proposed decision-making algorithm will correctly identify the degree of importance of each variable in influencing the output, and thus priority will be given to the variables that are deemed most important. This is then followed by the implementation of a regularization method that is custom-made for the prediction of diabetes. Such a customized regularization method is considered asymmetrical because the positive numbers are more favored compared to negative numbers, and this was decided based on the characteristics of the dataset. The proposed technique deals with missing numbers as a separate kind of entity compared to numerical entries and can adapt itself to a given dataset. The proposed customized hybrid model and its accompanying decision-making algorithm were applied to the Pima Indian Diabetes dataset sourced from the UCI Machine Learning Repository with an 80% prediction accuracy.

Keywords: Diabetes prediction; Disease prediction; Asymmetrical regularization; Artificial neural network; Genetic algorithm

1. Introduction

Diabetes mellitus is a metabolic disease caused by persistently high levels of glucose in the body and the body's inability to produce sufficient insulin to process this glucose. Both Types I and Type II diabetes alike continue to be an increasingly prominent disease throughout the world with Type II diabetes has reached epidemic proportions

* Corresponding author.
E-mail addresses: aghila25481@gmail.com (A. Rajagopal), jhasudan@ieee.org (S. Jha), ramautpc@gmail.com (R. Alagarsamy), queksg@ucsiuniversity.edu.my (S.G. Quek), Ganeshsree@ucsiuniversity.edu.my (G. Selvachandran).

https://doi.org/10.1016/j.matcom.2022.03.003

and has been declared a global pandemic by the WHO as far back as 2006. According to the American Diabetes Association, there were approximately 34.2 diabetics (about 1 in every 10 people) in the US as of mid-2020 and the WHO estimates that a whopping 463 million people around the world are suffering from diabetes as of 2020, and these are only the number of cases that have been diagnosed. Hence, diabetes studies such as diabetes prediction, and the study of the health complications caused by diabetes are of utmost importance.

The advent of big data and data analytics methods have had a huge impact in the areas of medicine and technology. The large amount of big data in medical research necessitates the use of frontier technologies such as machine learning, deep learning and cloud computing to fully utilize the big data and automate the computation processes in medical research, rather than using traditional approaches which are often unable to deal with big data. Hence, this paper proposes a customized hybrid model of artificial neural network (ANN) and genetic algorithms for an efficient diabetes disease prediction framework with regularization and prediction procedures that have been customized for the context of diabetes prediction.

Some of the recent studies related to diabetes prediction using data analytics and machine learning are expounded here. Razzak, Imran, and Xu [23] have provided a thorough overview of all the recent advancements of big data analytics for disease prevention, Tariq et al. [27] emphasized the importance of deep learning techniques for analyzing medical big data, while Wang and Alexander [28] highlighted the various challenges of medical big data analytics, such as the huge volume of data and the vast variety of formats. Lv and Qiao [21] had conducted a round of questionnaires to medical staff to assess the various risk factors and privacy issues associated with big data in healthcare. Both Lv and Qiao [21], as well as Li, Jiao, and Li [19] have outlined plenty of essential formulas to process medical big data.

Besides, in catering to the huge caliber of medical data, some recent works on the methods of data clustering (clustering in general) had surfaced in the literature. Li, Jiao, and Li [19] proposed a clustering method for heterogeneous data and then compared their proposed clustering method with some presented in previous literature using an experimental medical dataset. Alguliyev, Aliguliyev, and Sukhostat [1] introduced a method of weighted consensus clustering and tested its application on various real-life datasets, including one concerning diabetes. Besides, Bu et al. [4] introduced another method of clustering utilizing c-means algorithms. There were also two articles observed [15,25] which, unlike [1,4,19], are dedicated to the methodology of k-means clustering for medical big data. Shakeel et al. [25] developed a cloud-based framework for the diagnosis of diabetes using such technique of k-means clustering, whereas Khanmohammadi, Adibeig and Shanehbandy [15] developed a k-means clustering method for medical applications in general, hence it is potentially applicable to diabetes prediction too.

There were also some works done on introducing new methods dedicated to the prediction of diabetes through various means of data analytics. Zhu, Idemudia and Feng [35] improved the performance of a regression model in diabetes prediction by integrating k-means techniques and principal component analysis (PCA). Wu et al. [31] deployed k-means clustering which was said to achieve more than 95% accuracy for the real-life dataset that they used, while Lekha and Suchetha [18] deployed a convolutional neural network for a particular non-invasive detection device of diabetes. Dwivedi [8] analyzed several computing techniques used for diabetes prediction, Hassan et al. [11] predicted diabetes using an ensemble model that was developed by combining several machine learning algorithms, whereas Ramani, Devi, and Soundar [22], together with Zhou, Myrzashova, and Zheng [34] deployed a modified/enhanced structure of artificial neural networks for the detection of diabetic chronic disease. In addition, Battineni et al. [2] and Jayanthi, Babu and Rao [14] each had contributed a review article dedicated to the various prediction models for diabetes based on numerical data, while Wang, Tan, and Liu [29] reviewed the various deployment instants of particle swarm optimization (PSO) algorithms in the field of medicine. There have been two instants of works utilizing the multi-cascaded model, one by Hu et al. [12] for brain tumor prediction, the other by Shakeel, Karim, and Khan [26] for non-medical purposes.

Some of the recent studies related to the use of frontier technologies such as deep learning, cloud computing and artificial intelligence in the areas of medicine, medical diagnosis and precision health are expounded in this section. Zafar et al. [32] enhanced the issues of safety consideration connected with storage of cloud and highlighted the important schemes of data integrity for outsourced data. In this study, the taxonomy of existing data integrity schemes was presented to employ cloud storage. A relative investigation of present systems was also provided with a complete analysis of probable attacks of security with their mitigations. Fan et al. [9] proposed an identity-based protected combined signature (SIBAS) as a scheme of data integrity checking that resorts Trusted Execution Environment (TEE) as the auditor for checking the outsourced data on the local side. Not only can SIBAS check

the integrity of outsourced data, but it can also attain the management of security keys in TEE over Shamir's threshold system. Zhang et al. [33] projected a novel scheme of public verification for cloud storage with the use of indistinguishability complication which needs an inconsequential calculation on the auditor and the envoy most computation to the cloud. Imran et al. [13] addressed the data integrity problem in cloud computing by suggesting a system over which users can check the integrity of stored cloud data, and users can also track the occurrence of any data integrity violation. For this persistence, a new concept was utilized relatively in cloud computing termed "Provenance of Data". The check capabilities were important in cryptographic techniques and the flexibility of access control was increased through the proposed ABE method by Kumar et al. [20].

Onto-ACM was used to fix emerging cloud computing vulnerabilities by Choi, Choi, and Kim [6]. A stable computer audit protocol was suggested by Wei et al. [30] in which batch verification was performed to the safe storage and the sampling technique was optimized and the costs were reduced by the signature verified by the designer. A new patient-centered framework was proposed by Belguith et al. [3] to keep and view the online information, while Lei et al. [17] employed the Late Dirichlet Assignment (LDA) method to analyze the description of the service and explore the relation between the content and locational information. Ferrag et al. [10] reported the sample, the datasets used, and presented a comparative analysis of deep learning methods for information protection intrusion detection, while Devi et al. [7] applied the Modified Adaptive Neuro-Fuzzy Inference System (MANFIS) model for dynamic load balancing in a heterogeneous environment. [5] had developed a cloud computing framework utilizing 5G networks for the diagnosis of diabetes and the improvement of healthcare services, whereas Lahoura et al. [16] developed a cloud computing-based framework for breast cancer diagnosis.

There has been some deployment of machine learning algorithms without the involvement of any input datasets. One of such major contribution was accomplished by Samaniego et al. [24], for which machine learning is deployed alongside with partial differential equations to solve problems in computational mechanics. In such cases studies, deep neural networks were trained against conventional mechanical formulas in engineering in the attempt to replicate the results as closest possible yet saving the immense computational hassle of dealing with such conventional formulas, especially those involving symbolic integrations. In [24], the neural network structure was compared against 3 engineering systems all of which involved complicated formulas, namely cantilever nano-beam, simply supported square plate, and hollow sphere subjected to internal pressure. The results concluded that a deep neural network is indeed capable to produce results that are close to conventional formulas. This will potentially save significant computational resources, even in cases where formulas are available, albeit being very complicated.

On the other hand, it is worth mentioning that among all the references that were observed, all the machine learning algorithms deployed for the diagnosis of diabetes are supervised ones, and rely on real-life data as their inputs. This is done by constantly comparing the results yielded by the AI against a hard truth i.e., the results yielded by certified medical professionals. Such conventions are preserved because of the high risk involved in the field of medical sciences, in which patients' life and health are at stake. Furthermore, even in the conventional diagnosis of diabetes, conventional crisp mathematical formulas are completely absent. Such a situation is typical in the field of medicine, unlike most problems in engineering for which there are precise formulas and mathematical models.

The remainder of this paper is organized as follows. In Section 2, an overview of the structure of the Pima Indian Diabetes dataset that is used in this study is discussed. The proposed customized normalization method and the selection method for the input variables are introduced in Section 3, together with all the newly introduced formulas and prediction algorithm. In Section 4, the diabetes prediction results for the Pima Indian Diabetes dataset that is obtained from the implementation of our newly introduced customized hybrid model and its accompanying prediction algorithm are presented, analyzed, and discussed. Concluding remarks are presented in Section 5, followed by the acknowledgments and the list of references.

2. Structure of the dataset

This section delivers a detailed explanation of the proposed system. The overall flow of the proposed system is presented below.

2.1. Some essential descriptions of the raw data involved

In this article, the Pima Indians Diabetes dataset is used, which consists of 8 independent variables tested on 768 women of Pima Indian descendants. 500 of them are normal, the other 286 are diabetic.

V_1: Number of times the woman *became pregnant.*
V_2: Plasma glucose concentration at 2 h in an oral glucose tolerance test ($\times 0.1\,\text{mmol}/l$).
V_3: Blood pressure (mm Hg).
V_4: Triceps skinfold thickness (mm).
V_5: 2-h serum insulin $\left(\mu\text{IU/m}l \text{ or } \times \left(3.47 \times 10^{-8}\right) \text{mg/m}l\right)$.
V_6: Body mass index (BMI) $\left(\frac{\text{weight in kg}}{(\text{height inm})^2}\right)$
V_7: Diabetes pedigree function (a function which shows the likelihood of inheriting diabetes)
V_8: Age (years).

The dependent variable, Y, assigns a value of either 0 (negative) or 1 (positive).

It is therefore well understood in medical knowledge that, for each V_n aforementioned, the higher the value of V_n, generally the more likely $Y = 1$ will occur.

2.2. An overview of the dataset structure

The raw input consists of n variables $V_1, V_2, \ldots, V_n$. On m different instances, some (but may not be all) data values among the n input variables were taken simultaneously. This results in m collections of data value sets:

$$\left\{\left(\mathfrak{X}_{1,1}, \mathfrak{X}_{1,2}, \ldots, \mathfrak{X}_{1,n}\right), \left(\mathfrak{X}_{2,1}, \mathfrak{X}_{2,2}, \ldots, \mathfrak{X}_{2,n}\right), \ldots, \left(\mathfrak{X}_{m,1}, \mathfrak{X}_{m,2}, \ldots, \mathfrak{X}_{m,n}\right)\right\},$$

where $\mathfrak{X}_{i,j}$ is the set of data values (if it exists) of the variable V_j taken at the ith instance. In particular, if $\left|\mathfrak{X}_{i,j}\right| = 1$, we denote $\mathfrak{X}_{i,j} = \left\{x_{i,j}\right\}$.

Therefore, all raw input data values can be presented by a matrix:

$$\mathbf{A} = \begin{pmatrix} \mathfrak{X}_{1,1} & \mathfrak{X}_{1,2} & \cdots & \mathfrak{X}_{1,n} \\ \mathfrak{X}_{2,1} & \mathfrak{X}_{2,2} & & \mathfrak{X}_{2,n} \\ \vdots & & \ddots & \vdots \\ \mathfrak{X}_{m,1} & \mathfrak{X}_{m,2} & \cdots & \mathfrak{X}_{m,n} \end{pmatrix}.$$

Note that n is the number of variables involved, whereas m is the number of sets of data values. On the other hand, the actual output data values can be presented by a matrix $\mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$,

where y_i is the output from $\left(\mathfrak{X}_{i,1}, \mathfrak{X}_{i,2}, \ldots, \mathfrak{X}_{i,n}\right)$ for each i. Denote $X_j = \bigcup_{i=1}^{m} \mathfrak{X}_{i,j}$ for each j.

Remark. For the Pima Indians Diabetes dataset in particular, $n = 8$, $m = 768$, and $\left|\mathfrak{X}_{i,j}\right| \leq 1$ for all i and j. $\left|\mathfrak{X}_{i,j}\right| = 0$ even occurs for some i and j.

2.3. The aim of the computation technique

In the case of this study, the aim of the computation technique is to estimate y_i using $\left(\mathfrak{X}_{i,1}, \mathfrak{X}_{i,2}, \ldots, \mathfrak{X}_{i,n}\right)$ for all i.

2.4. A general visualization of the dataset structure

An illustration for the data distribution of $V_1, V_2, \ldots, V_n$ ($n = 8$ in the dataset used for this paper) is presented in Fig. 1. Here, the darker the color of a dot, the more the entries among the dataset having values at that dot. Such presentation is chosen to provide a detailed visualization of every single value which a variable may assign, which cannot be presented if the values are grouped together to form classes.

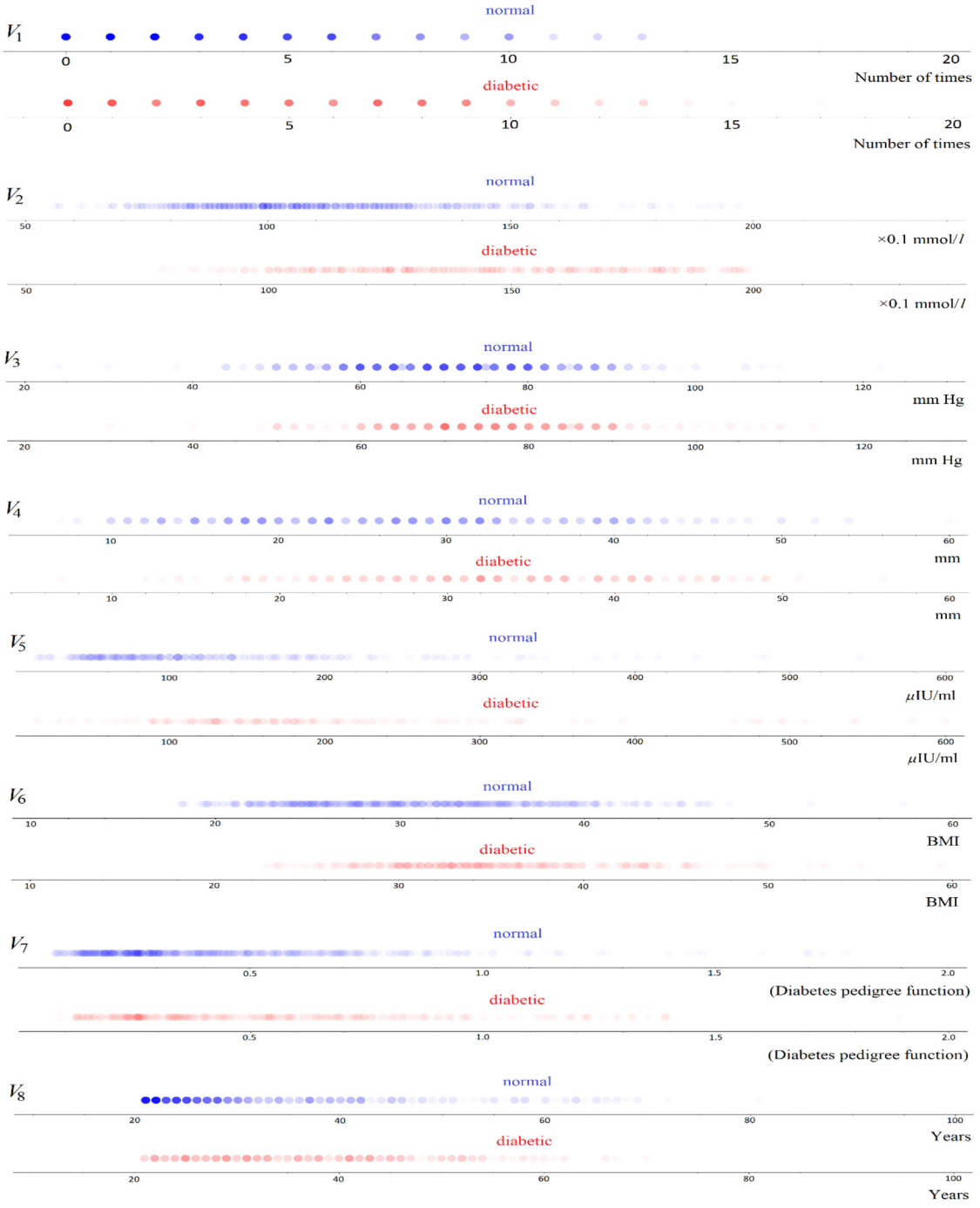

Fig. 1. A visualization of $V_1, V_2, \ldots, V_8$ showing the distribution pattern of the 8 independent variables.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. Proposed methodology

In this section, the newly introduced methodology of the customized normalization method and the selection method for the input variables, as well as all other details pertaining to the methodology proposed in this paper is presented.

3.1. Preprocessing using a novel way of normalization which works consistently for all degree of skewness

The set of normalized inputs are computed as follows.

$$\mathbf{B} = \begin{pmatrix} \hat{\mathfrak{x}}_{1,1} & \hat{\mathfrak{x}}_{1,2} & \cdots & \hat{\mathfrak{x}}_{1,n} \\ \hat{\mathfrak{x}}_{2,1} & \hat{\mathfrak{x}}_{2,2} & & \hat{\mathfrak{x}}_{2,n} \\ \vdots & & \ddots & \vdots \\ \hat{\mathfrak{x}}_{m,1} & \hat{\mathfrak{x}}_{m,2} & \cdots & \hat{\mathfrak{x}}_{m,n} \end{pmatrix}$$

Likewise, denote $\hat{X}_j = \bigcup_{u=1}^{m} \hat{\mathfrak{x}}_{u,j}$ for each j. For each i, j, the following holds:

If $\mathfrak{x}_{i,j} = \{x_{i,j}\}$, then $\hat{\mathfrak{x}}_{i,j} = \{\hat{x}_{i,j}\}$, where:

$$\hat{x}_{i,j} = \frac{x_{i,j} - \text{mode}\left(X_j\right)}{\left(2 \cdot \max\left\{\frac{\text{mode}\left(X_j\right) - \text{Q}_1\left(X_j\right)}{\text{Q}_3\left(X_j\right) - \text{Q}_1\left(X_j\right)}, 1 - \frac{\text{mode}\left(X_j\right) - \text{Q}_1\left(X_j\right)}{\text{Q}_3\left(X_j\right) - \text{Q}_1\left(X_j\right)}\right\}\right) \cdot \text{stdev}\left(X_j\right)}.$$

If $\mathfrak{x}_{i,j} = \varnothing$, then $\hat{\mathfrak{x}}_{i,j} = \varnothing$. Thus, in the formula above, if X_j is normally distributed, then:

$$\text{mean}\left(X_j\right) = \text{median}\left(X_j\right) = \text{mode}\left(X_j\right), 2\left(\text{median}\left(X_j\right) - \text{Q}_1\left(X_j\right)\right) = \text{Q}_3\left(X_j\right) - \text{Q}_1\left(X_j\right).$$

Then we have the following:

$$\begin{aligned} \hat{x}_{i,j} &= \frac{x_{i,j} - \text{mean}\left(X_j\right)}{\left(2 \cdot \max\left\{\frac{\text{median}\left(X_j\right) - \text{Q}_1\left(X_j\right)}{\text{Q}_3\left(X_j\right) - \text{Q}_1\left(X_j\right)}, 1 - \frac{\text{median}\left(X_j\right) - \text{Q}_1\left(X_j\right)}{\text{Q}_3\left(X_j\right) - \text{Q}_1\left(X_j\right)}\right\}\right) \cdot \text{stdev}\left(X_j\right)} \\ &= \frac{x_{i,j} - \text{mean}\left(X_j\right)}{\left(2 \cdot \max\left\{\frac{1}{2}, 1 - \frac{1}{2}\right\}\right) \cdot \text{stdev}\left(X_j\right)} = \frac{x_{i,j} - \text{mean}\left(X_j\right)}{\text{stdev}\left(X_j\right)}. \end{aligned}$$

Thus, $\hat{X}_j \sim \text{N}(0, 1)$.

On the other hand, if the distribution of $X_j - \varepsilon$ follows the negative exponential distribution for some $\varepsilon \in \mathbb{R}$, then minimum$\left(X_j\right) = \varepsilon$. Then $\hat{X}_j$ itself follows a negative exponential distribution. In such a case, we have the following:

$$\left(2 \cdot \max\left\{\frac{\text{median}\left(X_j\right) - \text{Q}_1\left(X_j\right)}{\text{Q}_3\left(X_j\right) - \text{Q}_1\left(X_j\right)}, 1 - \frac{\text{median}\left(X_j\right) - \text{Q}_1\left(X_j\right)}{\text{Q}_3\left(X_j\right) - \text{Q}_1\left(X_j\right)}\right\}\right) > 1$$

which brings all $\hat{x}_{i,j}$ closer to 0 to compensate the overall displacement of all $\hat{x}_{i,j}$ to the right side of 0 due to the very positive skewness of a negative exponential distribution.

The effectiveness of our normalization algorithm at handling different degrees of a distribution's skewness can even be observed in Fig. 2 with regard to the 8 input variables of the Pima Indian Diabetes Dataset. Particularly in Fig. 2, the blueish areas represent the 500 normal patients (i.e., $Y = 0$), whereas the reddish areas represent the other 286 diabetic patients (i.e., $Y = 1$).

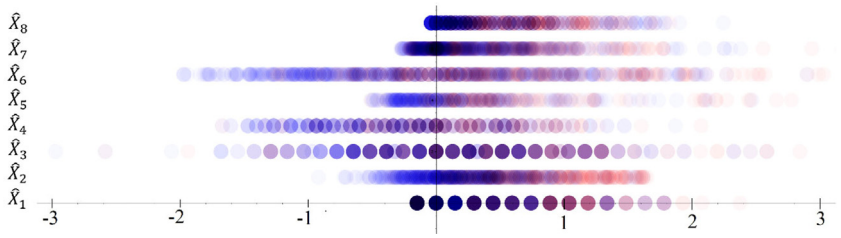

Fig. 2. The normalized data using our normalization technique, showing all values lined up properly and yet with their modes (represented by the darkest color) aligned near 0.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2. Selection of input variables that are most influential to the output variables, to be first considered during the subsequent machine learning process

Let $\Lambda = \{1 \le u \le m | y_u = 0\}$ and $V = \{1 \le u \le m | y_u = 1\}$. For each j, define

$$D_j = \frac{\text{mean}\left(\bigcup_{u\in V} \hat{\mathfrak{X}}_{u,j}\right) - \text{mean}\left(\bigcup_{u\in \Lambda} \hat{\mathfrak{X}}_{u,j}\right)}{\sqrt{\text{var}\left(\bigcup_{u\in V} \hat{\mathfrak{X}}_{u,j}\right) + \text{var}\left(\bigcup_{u\in \Lambda} \hat{\mathfrak{X}}_{u,j}\right)}}.$$

Let $\{\sigma(1), \sigma(2), \ldots, \sigma(n)\} = \{1, 2, \ldots, n\}$ be such that $D_{\sigma(h)} \ge D_{\sigma(k)}$ for all $h \le k$. In other words, the lower the value of i, the more influential (and hence the more important) the variable $V_{\sigma(i)}$.

Choose $r < n$, and then we form:

$$\mathbf{R}_a = \begin{pmatrix} \hat{\mathfrak{X}}_{1,\sigma(1)} & \hat{\mathfrak{X}}_{1,\sigma(2)} & \cdots & \hat{\mathfrak{X}}_{1,\sigma(r)} \\ \hat{\mathfrak{X}}_{2,\sigma(1)} & \hat{\mathfrak{X}}_{2,\sigma(2)} & & \hat{\mathfrak{X}}_{2,\sigma(r)} \\ & \vdots & \ddots & \vdots \\ \hat{\mathfrak{X}}_{m,\sigma(1)} & \hat{\mathfrak{X}}_{m,\sigma(2)} & \cdots & \hat{\mathfrak{X}}_{m,\sigma(r)} \end{pmatrix}, \mathbf{R}_b = \begin{pmatrix} \hat{\mathfrak{X}}_{1,\sigma(r+1)} & \hat{\mathfrak{X}}_{1,\sigma(r+2)} & \cdots & \hat{\mathfrak{X}}_{1,\sigma(n)} \\ \hat{\mathfrak{X}}_{2,\sigma(r+1)} & \hat{\mathfrak{X}}_{2,\sigma(r+2)} & & \hat{\mathfrak{X}}_{2,\sigma(n)} \\ & \vdots & \ddots & \vdots \\ \hat{\mathfrak{X}}_{m,\sigma(r+1)} & \hat{\mathfrak{X}}_{m,\sigma(r+2)} & \cdots & \hat{\mathfrak{X}}_{m,\sigma(n)} \end{pmatrix}$$

where $\mathbf{R}_a$ will be given attention first in the subsequent processes.

3.3. Detection of the most visible pattern on the most influential variables

The process of identifying the most visible patterns on the most important variables is presented in this section.

3.3.1. The inputs

By solely considering $\mathbf{R}_a$, form the input as follows:

$$\mathbf{P} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,r+1} \\ p_{2,1} & p_{2,2} & & p_{2,r+1} \\ \vdots & & \ddots & \vdots \\ p_{\eta,1} & p_{\eta,2} & \cdots & p_{\eta,r+1} \end{pmatrix} = \left(\begin{array}{c|ccc} x_{\vartheta(1),\sigma(1)} & x_{\vartheta(1),\sigma(1)} & x_{\vartheta(1),\sigma(2)} & \cdots & x_{\vartheta(1),\sigma(r)} \\ x_{\vartheta(2),\sigma(1)} & x_{\vartheta(2),\sigma(1)} & x_{\vartheta(2),\sigma(2)} & & x_{\vartheta(2),\sigma(r)} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{\vartheta(\eta),\sigma(1)} & x_{\vartheta(\eta),\sigma(1)} & x_{\vartheta(\eta),\sigma(2)} & \cdots & x_{\vartheta(\eta),\sigma(r)} \end{array}\right),$$

with

(i) $p_{i,1} = x_{\vartheta(i),\sigma(1)}$ for all $i \in \{1, 2, \ldots, \eta\}$.
(ii) $p_{i,1+j} = x_{\vartheta(i),\sigma(j)}$ for all $i \in \{1, 2, \ldots, \eta\}$ and $j \in \{1, 2, \ldots, r\}$.
(iii) $\{\vartheta(1), \vartheta(2), \ldots, \vartheta(\eta)\} \subseteq \{1, 2, \ldots, m\}$ are such that $\hat{\mathfrak{X}}_{\vartheta(i),\sigma(j)} = \left\{x_{\vartheta(i),\sigma(j)}\right\} \ne \varnothing$ for all $1 \le i \le \eta$ and $1 \le j \le r$.

Such definition grants another layer of priority to the most important variables of all, denoted as $V_{\sigma(i)}$ as its data values occupy two columns instead of one like the others.

3.3.2. Parameters to be subjected to machine learning

The set of parameters that will be used in Section 3.3.3 is as follows:

$$C = \left\{c_{i,j} | i \in \{1,2,\ldots,7\},\, j \in \{1,2,\ldots,r+1\}\right\} \cup \left\{\varsigma_{i,\{a,b\}} | i \in \{1,2,3\},\, \{a,b\} \subseteq \{1,2,\ldots,r+1\} \text{ with } a \neq b\right\}$$

3.3.3. The machine learning process for the detection of the most visible pattern

The steps involved in the detection of the most visible patterns are outlined below.

Step J1: Initialization

The machines learning process starts with the following initial choices:

(i) $c_{1,j} = 1$ for all j
(ii) $c_{i,j} = 0$ for all $i > 1$ and j
(iii) $\varsigma_{i,\{u,v\}} = 0$ for all i, u, v

Step J2: Determination of the estimated values

Calculate the following:

$$\begin{pmatrix} y'_{\vartheta(1)} \\ y'_{\vartheta(2)} \\ \vdots \\ y'_{\vartheta(\eta)} \end{pmatrix} = \sum_{j=1}^{r+1} c_{1,j} \begin{pmatrix} p_{1,j} \\ p_{2,j} \\ \vdots \\ p_{\eta,j} \end{pmatrix} + \sum_{j=1}^{r+1} c_{2,j} \begin{pmatrix} \left(p_{1,j} + c_{3,j}\right)^3 \\ \left(p_{2,j} + c_{3,j}\right)^3 \\ \vdots \\ \left(p_{\eta,j} + c_{3,j}\right)^3 \end{pmatrix}$$
$$+ \sum_{j=1}^{r+1} c_{4,j} \begin{pmatrix} \sqrt[3]{p_{1,j} + c_{5,j}} \\ \sqrt[3]{p_{2,j} + c_{5,j}} \\ \vdots \\ \sqrt[3]{p_{\eta,j} + c_{5,j}} \end{pmatrix} + \sum_{j=1}^{r+1} c_{6,j} \begin{pmatrix} e^{p_{1,j}} + c_{7,j} \\ e^{p_{2,j}} + c_{7,j} \\ \vdots \\ e^{p_{\eta,j}} + c_{7,j} \end{pmatrix}$$
$$+ \sum_{a=1}^{r} \sum_{b=a+1}^{r+1} \varsigma_{1,\{a,b\}} \begin{pmatrix} \max\left\{p_{1,a} + \varsigma_{2,\{a,b\}}, 0\right\} \cdot \max\left\{p_{1,b} + \varsigma_{3,\{a,b\}}, 0\right\} \\ \max\left\{p_{2,a} + \varsigma_{2,\{a,b\}}, 0\right\} \cdot \max\left\{p_{2,b} + \varsigma_{3,\{a,b\}}, 0\right\} \\ \vdots \\ \max\left\{p_{\eta,a} + \varsigma_{2,\{a,b\}}, 0\right\} \cdot \max\left\{p_{\eta,b} + \varsigma_{3,\{a,b\}}, 0\right\} \end{pmatrix}.$$

Step J3: Determination of the Area Under Curve (AUC)

Take:

(i) $Y'_0 = \left\{y'_{\vartheta(i)} | i = \{1,2,\ldots,\eta\} \text{ with } y_{\vartheta(i)} = 0\right\}$
(ii) $Y'_1 = \left\{y'_{\vartheta(i)} | i = \{1,2,\ldots,\eta\} \text{ with } y_{\vartheta(i)} = 1\right\}$

Find the Area Under Curve (AUC) generated by Y'_0 and Y'_1, and denoted as $\overline{\text{AUC}}_{Y'}$.

Step J4: Regularization

An asymmetrical regularization function customized for the scenario of diabetes prediction is deployed for our proposed technique, and is defined as follows: $\Psi = \psi_1 \sum_{j=1}^{r+1} \min\left\{0, c_{1,j}\right\} + \psi_2 \left(\sum_{i\in\{2,4,6\}} \sum_{j=1}^{r+1} \min\left\{0, c_{i,j}\right\} + \sum_{b=1}^{r} \sum_{b=a+1}^{r+1} \min\left\{0, \varsigma_{1,\{a,b\}}\right\}\right)$, where $\psi_1, \psi_2 \geq 0$ is a constant that is chosen as deemed appropriate for the context. Note that $\Psi \leq 0$. The score used in this machine learning process, $\overline{\text{SRE}}_0$, is henceforth defined as $\overline{\text{SRE}}_0 = \overline{\text{AUC}}_{Y'} + \Psi$.

Step J5: Iteration through genetic algorithm

Steps J2 and J4 of Section 3.3.3 are looped while deploying the genetic algorithm. As a result, Steps J1 to J5 of Section 3.3.3 will produce the following set of parameters: $\acute{C} = \left\{\acute{c}_{i,j} | i \in \{1, 2, \ldots, 7\}, j \in \{1, 2, \ldots, r+1\}\right\} \cup \left\{\acute{\varsigma}_{i,\{a,b\}} | i \in \{1, 2, 3\}, \{a, b\} \subseteq \{1, 2, \ldots, r+1\} \text{ with } a \neq b\right\}$ which gives the greatest $\overline{\text{SRE}}_0$ obtainable (not $\overline{\text{AUC}}_{Y'}$, for regularization matters). The set of parameters $\acute{C}$ is to be deployed in the final and the main process outlined in Section 3.5. It is worth noting that in Section 3.5, all variables are considered, whether they are considered important or not (see Step K1 of Section 3.5.3).

It is worth emphasizing that such priority of dealing with the important data in prior proves extremely important in real life analysis and it is able to reduce the complexity and time required for the computation process.

3.4. Processing of missing values

Define $\mathbf{Q}_a$ as follows to deal with the existing values from the important variables:

$$\mathbf{Q}_a = \begin{pmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,r+1} \\ q_{2,1} & q_{2,2} & & q_{2,r+1} \\ \vdots & & \ddots & \vdots \\ q_{m,1} & q_{m,2} & \cdots & q_{m,r+1} \end{pmatrix}$$

with:

(i) $q_{i,1} = x_{i,1}$ for all $i \in \{1, 2, \ldots, m\}$ with $\mathfrak{X}_{i,1} = \left\{x_{i,1}\right\}$.
(ii) $q_{i,1+j} = x_{i,j}$ for all $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, r\}$ with $\mathfrak{X}_{i,j} = \left\{x_{i,j}\right\}$.
(iii) $q_{i,j} = 0$ for all $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, r+1\}$ with $\mathfrak{X}_{i,j} = \{\}$.

Define $\mathbf{Q}_b$ as follows to deal with the existing values from the unimportant variables:

$$\mathbf{Q}_b = \begin{pmatrix} \acute{q}_{1,1} & \acute{q}_{1,2} & \cdots & \acute{q}_{1,n-r} \\ \acute{q}_{2,1} & \acute{q}_{2,2} & & \acute{q}_{2,n-r} \\ \vdots & & \ddots & \vdots \\ \acute{q}_{m,1} & \acute{q}_{m,2} & \cdots & \acute{q}_{m,n-r} \end{pmatrix}$$

with:

(i) $\acute{q}_{i,j} = x_{i,r+j}$ for all $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n-r\}$ with $\mathfrak{X}_{i,r+j} = \left\{x_{i,r+j}\right\}$.
(ii) $\acute{q}_{i,j} = 0$ for all $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n-r\}$ with $\mathfrak{X}_{i,j} = \{\}$.

Define $\mathbf{Q}_c$ as follows to deal with the missing values from all the variables:

$$\mathbf{Q}_c = \begin{pmatrix} \tilde{q}_{1,1} & \tilde{q}_{1,2} & \cdots & \tilde{q}_{1,n} \\ \tilde{q}_{2,1} & \tilde{q}_{2,2} & & \tilde{q}_{2,n} \\ \vdots & & \ddots & \vdots \\ \tilde{q}_{m,1} & \tilde{q}_{m,2} & \cdots & \tilde{q}_{m,n} \end{pmatrix}$$

with:

(i) $\tilde{q}_{i,j} = 0$ for all $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n\}$ with $\mathfrak{X}_{i,j} = \left\{x_{i,j}\right\}$.
(ii) $\tilde{q}_{i,j} = 1$ for all $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n\}$ with $\mathfrak{X}_{i,j} = \{\}$.

In this case, the entry of "1" does not mean that the missing value will be treated as 1, but only serves as a Boolean switch indicating that a value is missing.

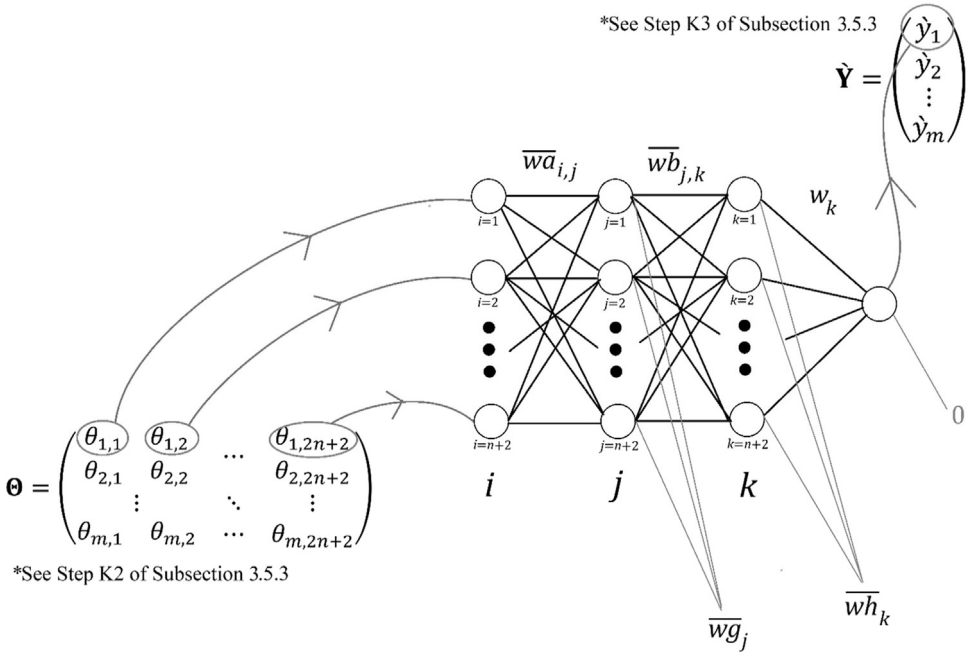

Fig. 3. The structure of our ANN model for diabetes prediction.

3.5. Finalizing the best solution using a hybrid model of artificial neural network, genetic algorithms, and a machine learning based preprocessing

The procedures of obtaining the final solution in the final part of the computation process are outlined in this section.

3.5.1. The inputs

The inputs consist of all the 3 matrices $\mathbf{Q}_a$, $\mathbf{Q}_b$ and $\mathbf{Q}_c$ that were obtained from Section 3.4.

3.5.2. Parameters to be subjected to machine learning

The set of parameters for the machine learning based preprocessing are denoted as follows: $D = \left\{d_{i,j} | i \in \{1, 2, \ldots, 6\}, j \in \{1, 2, \ldots, r+1\}\right\} \cup \left\{\delta_{i,\{a,b\}} | i \in \{1, 2, 3\}, \{a, b\} \subseteq \{1, 2, \ldots, r+1\} \text{ with } a \neq b\right\}$, whereas the set of parameters for the choices of dealing with missing values for each variable among $V_1, V_2, \ldots, V_n$ are denoted as $E = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$.

In all options of the neural network in our proposed technique, the number of input nodes is fixed to be $2n + 2$ (see Step K2 of Section 3.5.3), and the number of output nodes is fixed to be 1 (because AUC is deployed instead of accuracy).

On the other hand, in this demonstration, a 3-layer ANN is used, with $2n+2$ intermediate nodes in each of the 2 hidden layers, and 1 node for the output layer. The set of parameters for the ANN, i.e., the weights of the neurons in this case, are denoted as follows with reference to Fig. 3.

Denote the following in accordance with the ANN structure presented in Fig. 3:

$$\text{(i) } \overline{\mathbf{NA}} = \begin{pmatrix} \overline{wa}_{1,1} & \overline{wa}_{1,2} & & \overline{wa}_{1,2n+2} \\ \overline{wa}_{2,1} & \overline{wa}_{2,2} & \cdots & \overline{wa}_{2,2n+2} \\ \vdots & & \ddots & \vdots \\ \overline{wa}_{2n+2,1} & \overline{wa}_{2n+2,2} & \cdots & \overline{wa}_{2n+2,2n+2} \end{pmatrix}$$

(ii) $\overline{\mathbf{NB}} = \begin{pmatrix} \overline{wb}_{1,1} & \overline{wb}_{1,2} & \cdots & \overline{wb}_{1,2n+2} \\ \overline{wb}_{2,1} & \overline{wb}_{2,2} & & \overline{wb}_{2,2n+2} \\ \vdots & & \ddots & \vdots \\ \overline{wb}_{2n+2,1} & \overline{wb}_{2n+2,2} & \cdots & \overline{wb}_{2n+2,2n+2} \end{pmatrix}$

(iii) $\overline{\mathbf{NG}} = \begin{pmatrix} \overline{wg}_1 & \overline{wg}_2 & \cdots & \overline{wg}_{2n+2} \end{pmatrix}$

(iv) $\overline{\mathbf{NH}} = \begin{pmatrix} \overline{wh}_1 & \overline{wh}_2 & \cdots & \overline{wh}_{2n+2} \end{pmatrix}$

(v) $\mathbf{N} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{2n+2} \end{pmatrix}$

(vi) $\mathfrak{M} = \left\{ w | w \text{ is an element among anyone of } \overline{\mathbf{NA}}, \overline{\mathbf{NB}}, \overline{\mathbf{NG}}, \overline{\mathbf{NH}} \right\}$

(vii) $W = \{ w | w \text{ is an element of } \mathbf{N} \}$

As such, D, E, $\mathfrak{M}$ and W are the 4 sets containing all the parameters involved, all of which are subjected to machine learning, even including the parameters deployed in the preprocessing. As a result, the machine learning for the preprocessing which deals with missing values, and the neural network, will take place concurrently in our hybrid model.

3.5.3. The machine learning process for the finalizing hybrid model

The implementation procedures for the machine learning process to finalize the computation process of the hybrid model is outlined in this section.

Step K1: Initialization

At the beginning of the machine learning process, assign:

(i) $d_{i,j} = \acute{c}_{i+1,j}$ for all $i \in \{1, 2, \ldots, 6\}$ and $j \in \{1, 2, \ldots, r+1\}$.

(ii) $\delta_{i,\{a,b\}} = \acute{\varsigma}_{i,\{a,b\}}$ for all $i \in \{1, 2, 3\}$ and $\{a, b\} \subseteq \{1, 2, \ldots, r+1\}$ with $a \neq b$

(iii) $\varepsilon_j = 0$ for all $j \in \{1, 2, \ldots, n\}$.

(iv) $\overline{\mathbf{NA}} = \overline{\mathbf{NB}} = \mathbf{I}_{2n+2}$ which is the identity matrix of the appropriate dimension.

(v) $\overline{\mathbf{NG}} = \overline{\mathbf{NG}} = (1, 1, \ldots, 1)$

(vi) $\mathbf{N} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}\} r+2 \text{ terms} \\ \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}\} n-r \text{ terms} \\ \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}\} n \text{ terms}$, where the $n - r$ terms of 0 are to ensure that the unimportant variables are introduced at a slower rate.

Step K2: Preprocessing, which is also subjected to machine learning

Form the matrix $\mathbf{G} = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{pmatrix}$, where

$$\begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{pmatrix} = \sum_{j=1}^{r+1} d_{1,j} \begin{pmatrix} (q_{1,j} + d_{2,j})^3 \\ (q_{2,j} + d_{2,j})^3 \\ \vdots \\ (q_{m,j} + d_{2,j})^3 \end{pmatrix} + \sum_{j=1}^{r+1} d_{3,j} \begin{pmatrix} \sqrt[3]{q_{1,j} + d_{4,j}} \\ \sqrt[3]{q_{2,j} + d_{4,j}} \\ \vdots \\ \sqrt[3]{q_{m,j} + d_{4,j}} \end{pmatrix} + \sum_{j=1}^{r+1} d_{5,j} \begin{pmatrix} e^{q_{1,j}+d_{6,j}} \\ e^{q_{2,j}+d_{6,j}} \\ \vdots \\ e^{q_{m,j}+d_{6,j}} \end{pmatrix}$$
$$+ \sum_{a=1}^{r} \sum_{b=a+1}^{r+1} \delta_{1,\{a,b\}} \begin{pmatrix} \max\{q_{1,a} + \delta_{2,\{a,b\}}, 0\} \cdot \max\{q_{1,b} + \delta_{3,\{a,b\}}, 0\} \\ \max\{q_{2,a} + \delta_{2,\{a,b\}}, 0\} \cdot \max\{q_{2,b} + \delta_{3,\{a,b\}}, 0\} \\ \vdots \\ \max\{q_{m,a} + \delta_{2,\{a,b\}}, 0\} \cdot \max\{q_{m,b} + \delta_{3,\{a,b\}}, 0\} \end{pmatrix}.$$

Then, establish the preprocessed input:

$$\boldsymbol{\Theta} = \begin{pmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,2n+2} \\ \theta_{2,1} & \theta_{2,2} & & \theta_{2,2n+2} \\ \vdots & & \ddots & \vdots \\ \theta_{m,1} & \theta_{m,2} & \cdots & \theta_{m,2n+2} \end{pmatrix} = (\mathbf{Q}_a \,|\mathbf{G}|\, \mathbf{Q}_b | \mathbf{Q}_c)$$

which is the augmented matrix formed by co-joining $\mathbf{Q}_a$, $\mathbf{G}$, $\mathbf{Q}_b$ and $\mathbf{Q}_c$ in order from left to right. Note that $\boldsymbol{\Theta}$ has $2n + 2$ columns in total and $\boldsymbol{\Theta}$ will then be fed into the ANN.

Step K3: Determination of the estimated values with ANN

In accordance with the structure of the neural network presented in Fig. 3, the predicted values for the output, $\grave{\mathbf{Y}} = \begin{pmatrix} \grave{y}_1 \\ \grave{y}_2 \\ \vdots \\ \grave{y}_m \end{pmatrix}$ are calculated by $\grave{\mathbf{Y}} = \left(\left(\boldsymbol{\Theta}\overline{\mathbf{NA}} + \overline{\mathbf{NG}}\right)\overline{\mathbf{NB}} + \overline{\mathbf{NH}}\right)\mathbf{N}$.

Step K4: Determination of the AUC

Take:

(i) $\grave{Y}_0 = \left\{\grave{y}_i | i = \{1, 2, \ldots, m\} \text{ with } \grave{y}_i = 0\right\}$
(ii) $\grave{Y}_1 = \left\{\grave{y}_i | i = \{1, 2, \ldots, m\} \text{ with } \grave{y}_i = 1\right\}$

Find the AUC generated by $\grave{Y}_0$ and $\grave{Y}_1$, denoted as $\overline{\text{AUC}}_{\grave{Y}}$.

Step K5: Regularization

An asymmetrical regularization function that has been customized for the scenario of diabetes prediction is deployed for our proposed technique, and this function is defined as follows:

$$\Phi = \varphi_1 \sum_{j=1}^{r+1} \min\{0, d_{1,j}\} + \varphi_2 \left(\sum_{i\in\{3,5\}} \sum_{j=1}^{r+1} \min\{0, d_{i,j}\} + \sum_{b=1}^{r} \sum_{b=a+1}^{r+1} \min\{0, \delta_{1,\{a,b\}}\} \right)$$
$$- \varphi_3 \sum_{w\in\mathfrak{M}} (\min\{0, w\})^2 - \varphi_4 \sum_{w\in W} (\min\{0, w\})^2$$

where $\varphi_1, \varphi_2, \varphi_3, \varphi_4 \geq 0$ are constants chosen as appropriate. Note that $\Phi \leq 0$. The score used in this machine learning process, $\overline{\text{SRE}}$, is henceforth defined as $\overline{\text{SRE}} = \overline{\text{AUC}}_{\hat{Y}} + \Phi$.

Step K6: Iteration through genetic algorithm

Steps K2 and K5 outlined in Section 3.5.3 are looped while deploying the genetic algorithm until the desired benchmark of AUC is finally met/achieved.

3.5.4. Training and testing

A 6-fold nested cross validation procedure was deployed for the data. As there are 768 sets of variables, the cross validation were conducted as follows.

Session number	Sets of variables for training	Sets of variables for testing
1	1–640	641–768
2	1–512, 641–768	513–640
3	1–384, 513–768	385–512
4	1–256, 385–768	257–384
5	1–128, 257–768	129–256
6	129–768	1–128

In each session, only the sets of training variables were involved in the machine learning process, whereas the values of AUC were computed only using the sets of testing variables.

The resultant weights of the neural networks (see Section 4.2.3), the AUCs (see Section 4.2.1), as well as all the formulas deduced by the program (see Section 4.2.2), are thus taken to be the average among the 6 sessions. The final results are presented in Section 4.

4. Discussion and analysis of results

In this section, the results obtained from the implementation of our newly introduced customized hybrid model and its accompanying decision-making algorithm to the Pima Indian Diabetes dataset that were sourced from the UCI Machine Learning Repository is presented. A comprehensive analysis and discussion of the results are then presented.

4.1. The performance of our proposed technique even under regularization

During testing, all combinations of $(\psi_1, \psi_2, \varphi_1, \varphi_2, \varphi_3, \varphi_4)$ below were tested, each on many different seeds of random numbers:

$\varphi_1, \varphi_2 \in \{1, 10, 100, 1000, 10000\}$, $(\psi_1, \psi_2) = (\varphi_1, \varphi_2)$, $\varphi_3 = 10$, $\varphi_4 = 200$.

Such a wide range of values from 1 to 10000 represents a vastly different degree of strictness as to whether some unusual relationships may be tolerated in the computations. Nonetheless, AUC = 81.0±0.5% was achieved under all trials, where each trial was run on one CPU core for 2 h or until the computations were halted by the program. Note that such AUC was achieved even under very strict regularization where all values of $\psi_1, \psi_2, \varphi_1, \varphi_2, \varphi_3, \varphi_4$ are extremely large, thus strictly forbidding neurons of negative weights. Such stringent choice of regularizations ensures that only reasonable values and parameters are considered throughout the entire procedure. Moreover, it was unanimously concluded by all choices of $(\psi_1, \psi_2, \varphi_1, \varphi_2, \varphi_3, \varphi_4)$ observed that glucose level (V_2) has the highest effect on the risk of diabetes, followed by BMI (V_6), age (V_8), and lastly, the number of pregnancies (V_1).

In contrast, all the previous studies related to diabetes prediction in literature did not deploy any regularization function to arrive at their results, and this is evident in all their programming codes that are available to download on the Internet or available in the appendices of their articles. In such a situation, severe overfitting often occurs, rendering these algorithms unable to perform in real-life applications even if such algorithms may have achieved very high AUC values for the data values provided.

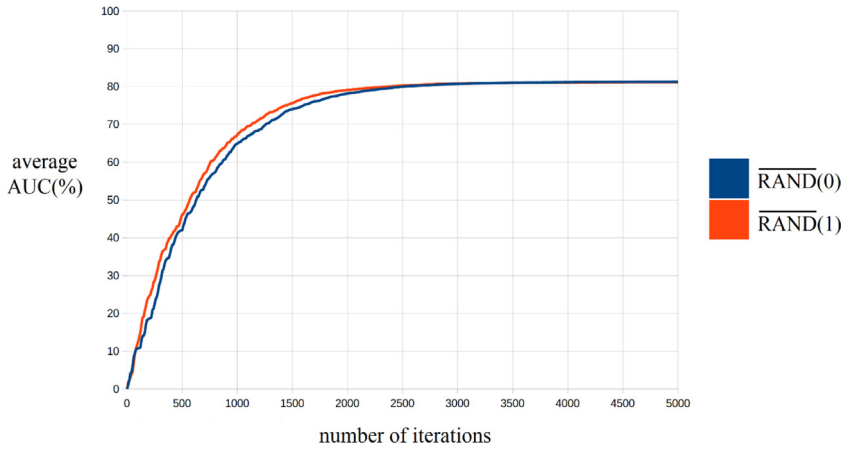

Fig. 4. The learning curve for $\overline{\text{RAND}}(0)$ and $\overline{\text{RAND}}(1)$.

Another advantage of our study is that we have utilized the entire available dataset, and we do not reject any among the 768 sets of variables presented in the datasets in our computations, no matter the quality of the data. As a result, our proposed algorithm can even work well under less-than-ideal conditions when many data have reached abnormal values. This in contrast to some of the other studies in the existing literature who claimed to have achieved an AUC of above 90.0%, had rejected many sets (in some studies even more than 100 out of the 768 available data were rejected and not used in the computation process) of input variables for various reasons, hence their proposed algorithms are only capable of dealing with very ideal data and not able to perform under less than ideal conditions with less than ideal (e.g. incomplete and abnormal) data.

4.2. Result analytics: An overview

Due to the limitation of scope, we consider the results obtained from

$$(\psi_1, \psi_2, \varphi_1, \varphi_2, \varphi_3, \varphi_4) = (1000, 10, 1000, 10, 10, 200)\,,$$

using two seeds of random numbers $\overline{\text{RAND}}(0)$ and $\overline{\text{RAND}}(1)$ available on SAGE, which are just two instances of all the combinations of $\psi_1, \psi_2, \varphi_1, \varphi_2, \varphi_3, \varphi_4$ that we have tested, as highlighted in Section 4.1.

4.2.1. AUC for the two examples

From $(\psi_1, \psi_2, \varphi_1, \varphi_2, \varphi_3, \varphi_4) = (1000, 10, 1000, 10, 10, 200)$, and across all 6 nested validations, an average AUC of 81.3% and 81.2% was obtained using $\overline{\text{RAND}}(0)$ and $\overline{\text{RAND}}\,(1)$, respectively. The learning curves are as presented in Fig. 4.

4.2.2. Formulas for preprocessing

From $(\psi_1, \psi_2, \varphi_1, \varphi_2, \varphi_3, \varphi_4) = (1000, 10, 1000, 10, 10, 200)$, it was deduced that:

$$\begin{aligned} g_i = {} & \left(0.22446x_{i,6}^3 - 0.51997x_{i,6}^2 - 5.83402x_{i,6}\right) + 0.02920x_{i,1}^3 + 0.84954x_{i,2}^2 + 6.48594x_{i,2}x_{i,6} \\ & + 0.02478x_{i,1}^2 + 16.42674x_{i,2} + 0.00701x_{i,1} + 0.50059\left(x_{i,2} + 0.12607\right)^{\frac{1}{3}} \\ & + 0.27265(x_{i,2} - 0.00557)^{\frac{1}{3}} + 0.61670(x_{i,8} - 0.25536)^{\frac{1}{3}} - 17.89097 \end{aligned}$$

for all $1 \le i \le m$ after $\overline{\text{RAND}}(0)$ is used.

On the other hand, it was deduced that:

$$\begin{aligned} g_i = {} & \left(1.65052x_{i,2}^3 - 1.50018x_{i,2}^2 + 7.50819x_{i,2}\right) + \left(0.18864x_{i,6}^3 - 0.30648x_{i,6}^2 - 1.68293x_{i,6}\right) \\ & + 0.04162x_{i,2}x_{i,8} + 0.60649x_{i,1}x_{i,2} + 0.06067x_{i,1}x_{i,6} + 2.08319x_{i,2}x_{i,6} \end{aligned}$$

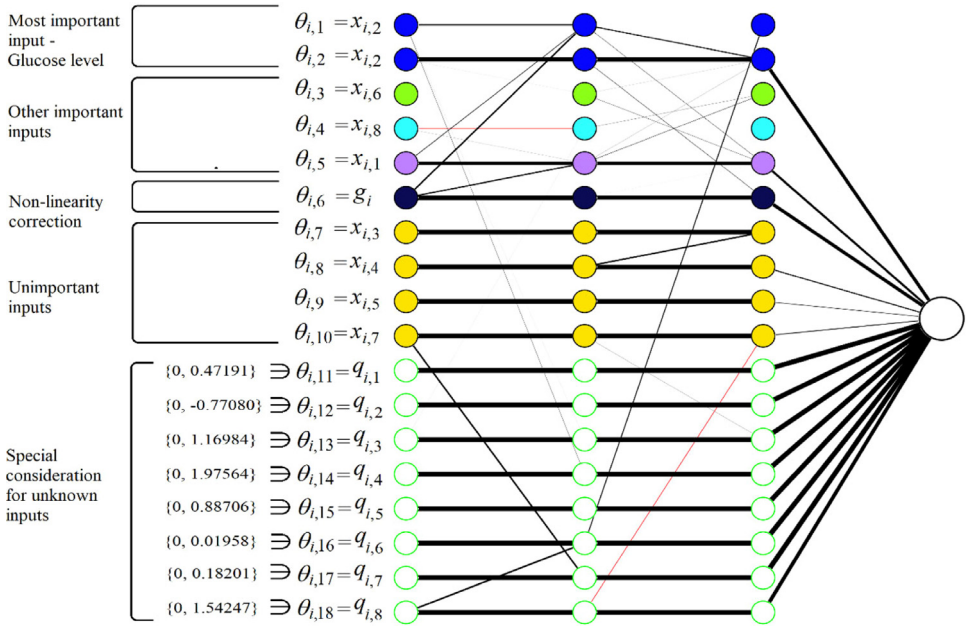

Fig. 5. The structure of the neural network obtained using $\overline{\text{RAND}}(0)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$+ 0.01698x_{i,8} + 0.28964x_{i,1} + 1.25922\left(x_{i,2} + 0.40217\right)^{\frac{1}{3}} + 1.57733\left(x_{i,2} + 0.38552\right)^{\frac{1}{3}}$$
$$+ 2.91731\left(x_{i,6} + 0.82575\right)^{\frac{1}{3}} + 1.82845\left(x_{i,8} - 0.25928\right)^{\frac{1}{3}} + 0.19278\mathrm{e}^{x_{i,2}-0.21508}$$
$$- 6.73544$$

for all $1 \leq i \leq m$ after $\overline{\text{RAND}}(1)$ is used.

As can be seen in the formulas above, the coefficients are mostly positive, which is consistent with the nature of the datasets, as mentioned in Section 2.1. In particular, for $x_{i,6}$ which represents the BMI, the highest power $x_{i,6}^3$ remains positive, which indicates that once glucose levels pass a certain threshold value, the risk of having diabetes will begin to increase rapidly.

4.2.3. Structures of the neural network

The structure of the neural network that is yielded when $\overline{\text{RAND}}(0)$ and $\overline{\text{RAND}}(1)$ are used on $(\psi_1, \psi_2, \varphi_1, \varphi_2, \varphi_3, \varphi_4) = (1000, 10, 1000, 10, 10, 200)$ is as shown in Figs. 5 and 6, respectively.

In particular, the black lines and red lines in Figs. 5 and 6 represent neurons with positive weight and negative weight, respectively. Moreover, the width of the line drawn in Figs. 5 and 6 corresponds to the weight magnitude of the neurons. Hence, the higher the weight magnitude of the neurons, the wider the line. For example, a neuron with the weight of 1.0 will be displayed as a wider line compared to a neuron with the weight of 0.1, but both lines are nonetheless presented in black color as both represent positive weights.

In both Figs. 5 and 6, it can be concluded that:

(i) The greater the 8 raw inputs as mentioned in Section 2.1, generally the higher the risk of diabetes, evident from the great abundance of neurons with positive weights over the few neurons whose weights are negative.
(ii) Glucose levels (V_2) have the highest effect on the risk of diabetes, followed by BMI (V_6), age (V_8), and lastly the number of pregnancies (V_1).
(iii) The values of $\overline{wg}_j$ and $\overline{wh}_k$ (see Section 3.5.2) are kept as zero for all j and k, showing that the preprocessing which is also machine learning based has already resolved most of the non-linear relationships between the variables, leaving the neural network to do the final corrections.

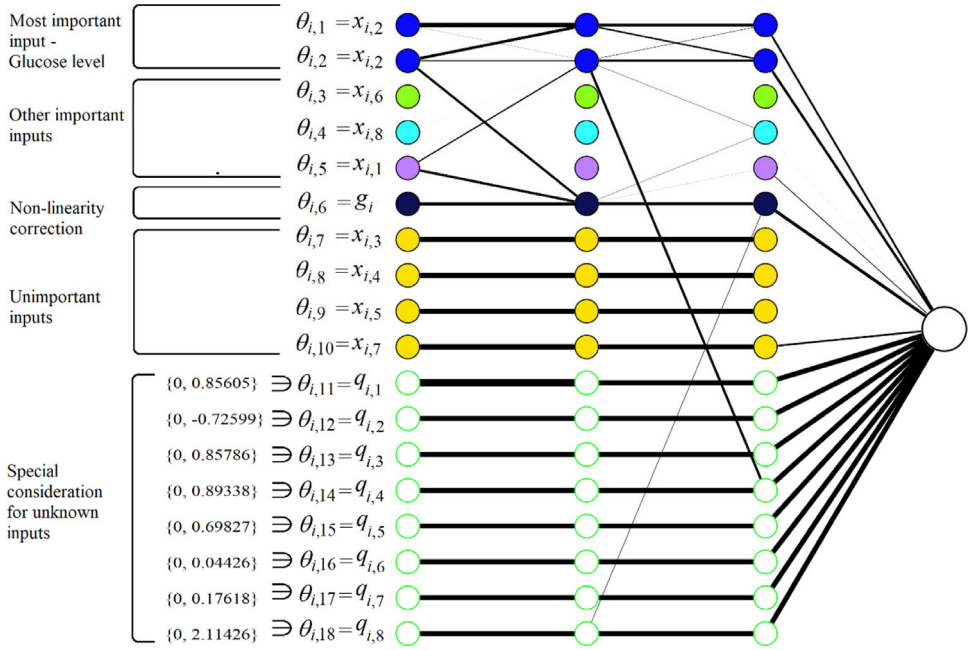

Fig. 6. The structure of the neural network obtained using $\overline{\text{RAND}}(1)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(iv) The variables V_3, V_4, V_5, V_7 are confirmed to have little or even no effect on the output, evident from the thin line or even the absence of lines joining the second intermediate layer with the output.
(v) For the treatment of unknown values, $q_{i,2}$ (absence of data for glucose) should be assigned a negative number, $q_{i,6}$ (absence of data for BMI) should be slightly positive, whereas $q_{i,8}$ (absence of data for age) should be the most positive.

4.3. Novelty of the proposed decision-making algorithm

The proposed prediction algorithm in this paper was proven to be able to overcome the limitations of the existing techniques and provides an effective framework for the detection and prediction of diabetes. The major novelty of the techniques proposed in this paper, in the order of deployment during the computation algorithm, are as follows:

(i) To prevent overfitting, the regularization function adopted throughout the proposed technique is also customized following the influence of all V_n on Y.
(ii) The proposed technique is a unique normalization method that adapts itself with the degree of skewness of each variable's distribution. This enables our normalization method to work simultaneously on many heterogeneous variables, ranging from the most symmetrical (e.g., a normal distribution) to the most skewed (a negative exponential distribution), yet the normalized values remain centered at the appropriate positions.
(iii) The proposed techniques identify potential missing values on their own, even if the entries were merely presented as "0" in the raw data. The entries with missing values are treated uniquely compared to the entries whose values are known, and the AI can even decide on the best way of enabling entries with missing values to be used in the computations.
(iv) The proposed techniques can identify the most influential input variable in determining the output. During the computations, variables that are deemed the most important will be given priority and thus processed earlier compared to the variables that are deemed less important. This matches the inherent nature of medical diagnosis in which all the risk factors contributing to a disease often have different degrees of severity.
(v) Before deploying a neural network, the proposed technique identifies the more visible non-linear patterns using some common equations, whose parameters are acquired through the implementation of genetic algorithms.

5. Conclusion

The main contributions and significant findings of this study are summarized below.

(i) Among all the studies that have used the Pima Indians Diabetes dataset, our work is among the ones which truly address the context of diabetes, in which all the steps of our algorithms, such as the regularization function, are dedicatedly customized for the context of diabetes. Positive values are favored over negative values as per the human understanding and knowledge of diabetes. This is because, in all the 8 criteria that were given in the dataset, it is well understood that the higher the criteria's value (e.g., blood glucose levels). The more prone the person is to diabetes. This is a sharp contrast to all the previous studies in this area that used readily available, general-purpose packages which are unable to specifically address the scenario of diabetes prediction.

(ii) Our work has also revealed a very important characteristic of the Pima Indians Diabetes Dataset: many of its entries are absent, not zero. Such absence of data requires a unique way of handling the data which was very rarely mentioned (if any) in all the previous works in the literature concerning the Pima Indians Diabetes Dataset or any other datasets with similar characteristics. Although the absence of data in the Pima Indians Diabetes Dataset was also represented as the number "0" in this study, it is easily distinguishable by a human. For example, since it is well known that a person's BMI cannot reach 0, if a data of BMI as 0 shows up, it shall be taken to mean the absence of data as is appropriate in this context. In addition, our program will even decide on its own the best way to handle missing values for a given input variable.

(iii) Unlike many studies even on real-life datasets, the AI algorithms that have been introduced in this study will first examine the importance of each input variable in influencing the final result, and therefore, priority will be given to the variables that are of the most importance. This is a faithful simulation of human intuition and knowledge, especially in medical diagnosis. In the case of diabetes prediction, it was determined by the AI that a diabetes diagnosis result, whether positive or negative is most significantly determined by blood sugar levels, followed by BMI (obesity), age, and the number of times being pregnant. Such a conclusion matches perfectly with the understanding of human experts (i.e., doctors) in the diagnosis of diabetes mellitus.

(iv) It is also understood that many other factors, such as the lack of exercise, are also factors that contribute to a higher risk or prevalence of diabetes. Such additional data are however absent from the Pima Indians Diabetes dataset. This is in addition to many missing values within the 768 sets of inputs. Yet despite working with such incomplete collections of data, our study still managed to achieve an accuracy of 80%–81% matching with the hard truth i.e. the diagnosis made by doctors.

(v) Our proposed decision-making algorithm is entirely custom-made for a given scenario, in this case for the prediction of diabetes, and we did not deploy any readily available packages in our computations. Therefore, compared to the previous studies in this area, our proposed decision-making technique has a higher level of novelty and is more suited for diabetes prediction in the real world.

Funding

This research was funded by the Ministry of Education, Malaysia under grant no. FRGS/1/2020/STG06/UCSI/02/1.

CRediT authorship contribution statement

Aghila Rajagopal: Conceptualization, Resources, Writing – original draft. **Sudan Jha:** Conceptualization, Resources, Writing – second draft, Supervision. **Ramachandran Alagarsamy:** Conceptualization, Writing – original draft, Investigation. **Shio Gai Quek:** Methodology, Data curation, Writing – second draft, Visualization, Formal analysis, Investigation, Validation, Resources, Software. **Ganeshsree Selvachandran:** Methodology, Formal analysis, Investigation, Validation, Writing – review & editing, Funding acquisition, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the Editor-in-Chief and the anonymous reviewers for their valuable comments and suggestions.

References

[1] R.M. Alguliyev, R.M. Aliguliyev, L.V. Sukhostat, Weighted consensus clustering and its application to big data, Expert Syst. Appl. 150 (2020) 113294.
[2] G. Battineni, G.G. Sagaro, N. Chinatalapudi, F. Amenta, Applications of machine learning predictive models in the chronic disease diagnosis, J. Pers. Med. 10 (21) (2020).
[3] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, R. Attia, Phoabe: Securely outsourcing multi-authority attribute-based encryption with policy hidden for cloud assisted IOT, Comput. Netw. 133 (2018) 141–156.
[4] F. Bu, C. Hu, Q. Zhang, C. Bai, L.T. Yang, T. Baker, A cloud–edge-aided incremental high-order possibilistic c-means algorithm for medical data clustering, IEEE Trans. Fuzzy Syst. 29 (1) (2020) 148–155.
[5] M. Chen, J. Yang, J. Zhou, Y. Hao, J. Zhang, C. Youn, 5G-smart diabetes: Toward personalized diabetes diagnosis with healthcare big data clouds, IEEE Commun. Mag. 56 (4) (2018) 16–23.
[6] C. Choi, J. Choi, P. Kim, Ontology-based access control model for security policy reasoning in cloud computing, J. Supercomput. 67 (2014) 711–722.
[7] T.D. Devi, A. Subramani, P. Anitha, Modified adaptive neuro fuzzy inference system-based load balancing for virtual machine with security in cloud computing environment, J. Ambient Intell. Humaniz. Comput. 12 (2021) 3869–3876.
[8] A.K. Dwivedi, Analysis of computational intelligence techniques for diabetes mellitus prediction, Neural Comput. Appl. 30 (2018) 3837–3845.
[9] Y. Fan, X. Lin, G. Tan, Y. Zhang, W. Dong, J. Lei, One secure data integrity verification scheme for cloud storage, Future Gener. Comput. Syst. 96 (2019) 376–385.
[10] M.A. Ferrag, L. Maglaras, S. Moschoyiannis, H. Janicke, Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study, J. Inf. Secur. Appl. 50 (2020) 102419.
[11] M.K. Hassan, M.A. Alam, D. Das, E. Hossain, M. Hasan, Diabetes prediction using ensembling of different machine learning classifier, IEEE Access 8 (2020) 76516–76531.
[12] K. Hu, Q. Gan, Y. Zhang, S. Deng, F. Xiao, W. Huang, C. Cao, X. Gao, Brain tumor segmentation using multi-cascaded convolutional neural networks and conditional random field, IEEE Access 7 (2019) 92615–92629.
[13] M. Imran, H. Hlavacs, B.J. Inam Ul Haq, F.A. Khan, A. Ahmad, Provenance based data integrity checking and verification in cloud environments, PLoS One 12 (5) (2017) e0177576.
[14] N. Jayanthi, B.V. Babu, N.S. Rao, Survey on clinical prediction models for diabetes prediction, J. Big Data 4 (26) (2017).
[15] S. Khanmohammadi, N. Adibeig, S. Shanehbandy, An improved overlapping k-means clustering method for medical applications, Expert Syst. Appl. 67 (2017) 12–18.
[16] V. Lahoura, H. Singh, A. Aggarwal, B. Sharma, M.A. Mohammed, R. Damaševičius, S. Kadry, K. Cengiz, Cloud computing-based framework for breast cancer diagnosis using extreme learning machine, Diagnostics 11 (241) (2021).
[17] C. Lei, H. Dai, Z. Yu, R. Li, A service recommendation algorithm with the transfer learning-based matrix factorization to improve cloud security, Inform. Sci. 513 (2020) 98–111.
[18] S. Lekha, M. Suchetha, Real-time non-invasive detection and classification of diabetes using modified convolution neural network, IEEE J. Biomed. Health Inf. 22 (5) (2018) 1630–1636.
[19] X. Li, H. Jiao, D. Li, Intelligent medical heterogeneous big data set balanced clustering using deep learning, Pattern Recognit. Lett. 138 (2020) 548–555.
[20] M. Li, S. Yu, Y. Zheng, K. Ren, W. Lou, Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption, IEEE Trans. Parallel Distrib. Syst. 24 (1) (2013) 131–143.
[21] Z. Lv, L. Qiao, Analysis of health care big data, Future Gener. Comput. Syst. 109 (2020) 103–110.
[22] R. Ramani, K.V. Devi, K.R. Soundar, MapReduce-based big data framework using modified artificial neural network classifier for diabetic chronic disease prediction, Soft Comput. 24 (2020) 16335–16345.
[23] M.I. Razzak, M. Imran, G. Xu, Big data analytics for preventive medicine, Neural Comput. Appl. 32 (2020) 4417–4451.
[24] E. Samaniego, C. Anitescu, S. Goswami, V.M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, T. Rabezuk, An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications, Comput. Methods Appl. Mech. Eng. 362 (2020) 112790.
[25] P.M. Shakeel, S. Baskar, V.R.S. Dhulipala, M.M. Jaber, Cloud based framework for diagnosis of diabetes mellitus using K-means clustering, Health Inf. Sci. Syst. 6 (16) (2018).
[26] M.H. Shakeel, A. Karim, I. Khan, A multi-cascaded model with data augmentation for enhanced paraphrase detection in short texts, Inf. Process. Manage. 57 (2020) 102204.
[27] M.I. Tariq, S. Tayyaba, M.W. Ashraf, V.E. Balas, Deep learning techniques for optimizing medical big data, Deep Learn. Tech. Biomed. Health Inform. 2020 (2020) 187–211.
[28] L. Wang, C.A. Alexander, Big data analytics in medical engineering and healthcare: Methods, advances and challenges, J. Med. Eng. Technol. 44 (6) (2020) 267–283.
[29] D. Wang, D. Tan, L. Liu, Particle swarm optimization algorithm: An overview, Soft Comput. 22 (2018) 387–408.

[30] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, A.V. Vasilakos, Security and privacy for storage and computation in cloud computing, Inform. Sci. 258 (2014) 371–386.
[31] H. Wu, S. Yang, Z. Huang, J. He, X. Wang, Type 2 diabetes mellitus prediction model based on data mining, Inform. Med. Unlocked 10 (2018) 100–107.
[32] F. Zafar, A. Khan, S.U.R. Malik, M. Ahmed, A. Anjum, M.I. Khan, N. Javed, M. Alam, F. Jamil, A survey of cloud computing data integrity schemes: Design challenges, taxonomy and future trends, Comput. Secur. 65 (2017) 29–49.
[33] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu, X. Zhang, Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation, IEEE Trans. Inf. Forensics Secur. 12 (2016) 676–688.
[34] H. Zhou, R. Myrzashova, R. Zheng, Diabetes prediction model based on an enhanced deep neural network, EURASIP J. Wireless Commun. Networking 2020 (148) (2020).
[35] C. Zhu, C.U. Idemudia, W. Feng, Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques, Inf. Med. Unlocked 17 (2019) 100179.