DOI: 10.37190/abb/209528

A Novel Data Mining Approach for Early Diagnosis of Gestational Diabetes Mellitus (GDM) in Pregnancy via Machine Learning Methods and CNN

Muhammet Serdar Başçıl[1*]

[1]Department of Biomedical Engineering, Faculty of Technology, Selcuk University, Konya, Türkiye

[*]Corresponding author: Muhammet Serdar Başçıl, Department of Biomedical Engineering, Faculty of Technology, Selcuk University, Konya,Türkiye, e-mail address: serdar.bascil@selcuk.edu.tr

Submitted: 11th June 2025

Accepted: 14th August 2025

Abstract

Purpose: The aim of this study was to investigate a novel data mining approach for early and effective diagnosis of Gestational Diabetes Mellitus (GDM).

Methods: Gestational Diabetes Mellitus (GDM) data contains two classes (healthy and diabetic), 15 features and 3525 instances. In the first stage, the widely used and effective KNN and Regression methods were employed for the filling of missing data. Then, the data source transformed into grayscale images as primary images and multiplexed images. Finally, both original data and transformed data are classified with KNN, SVM and CNN using k-fold cross validation technique. Performance metrics were compared to extract the best suitable system.

Results: The original GDM source and the missing values replacement of GDM are classified with KNN and SVM methods. Also, primary images of this dataset and multiplexed images are classified with CNN 50%-50% and 70%-30% train-test respectively. The results of classification performance demonstrated that reaching up to 97,91% with CNN, recall of 97,61%, specificity of 97,61%, Precision of 97,97%, and F1-score of 97,79%. This result outperformed all previous studies conducted on the same dataset in the literature.

Conclusion: This work is demonstrated a new approach that the best results of classification accuracy when compared with previous studies related to proposed methods to identify GDM disease. It can be clearly stated that applying a data mining method to impute missing values, followed by converting the dataset into images based on certain criteria and classifying with CNN, is the most effective approach for predicting GDM.

Keywords: GDM disease, Data mining, Image conversion, KNN, SVM, CNN.

1. Introduction

Diabetes is classified into three known types named as Type 1 Diabetes (T1D), Type 2 Diabetes (T2D), and Gestational Diabetes Mellitus (GMM) [1],[11]. GDM is one of a prevalent disease that occurs during pregnancy and cause problems for both the mother and the baby. If it is not controlled and treated, it causes a long term cardiovascular and neuro complications for mothers and infants. Fort this reason, GDM is a growing public health concern [8], [25]. After birth, the mother may have the possibility of T1D or T2D. Various risk factors have been identified for gestational diabetes mellitus (GDM), including a previous occurrence of the condition, a family history of type 2 diabetes, ethnicity, older maternal age, lifestyle choices, and dietary habits. Additionally, psychosocial and environmental influences, such as exposure to endocrine disruptors, organic pollutants, and experiencing depression during the first and second trimesters, have been suggested as potential contributors to GDM development. Genetic predisposition may also play a role in the progression of GDM, although existing research remains inconclusive and inconsistent [11], [25]. The infant may experience the problem of poor nutrition and be prone to diabetes in the future. Managing blood glucose levels in the treatment of GDM can be achieved through various approaches, including maintaining a healthy diet, engaging in regular physical activity, and using medication such as oral tablets or insulin injections [3], [37].

The conventional approach to early intervention involves doctors assessing the likelihood of disease occurrence based on patient's basic information and personal experience, including demographics, existing medical conditions, and lifestyle habits, before implementing preventive measures. However, this method often lacks high accuracy and can be influenced by subjective judgment. With advancements in information technology, hardware improvements, and the emergence of new theoretical frameworks, disease prediction has become increasingly precise through various predictive techniques. Among these, machine learning is widely utilized as an effective tool. It plays a key role in processing large-scale data and is extensively applied across multiple fields [21]. With advancements in modern technology, vast amounts of data are continuously collected, enabling the effective use of machine learning in healthcare. Physicians can assess a patient's condition using clinical metrics such as blood pressure and body temperature, allowing for more precise treatment planning through iterative analysis and refinement. Additionally, artificial intelligence plays a crucial role in disease classification and diagnosis [24]. While challenges exist, particularly with computer-aided interpretation, deep learning techniques are increasingly being employed to enhance diagnostic accuracy and improve patient outcomes [16], [31], [43].

In the literature, there are studies conducted on different GDM datasets. The highlights of these studies can be summarized as follows: Shen et al. aimed to diagnose GDM using only the patient's age and fasting blood glucose values in regions with limited medical resources [32]. Nine different machine learning algorithms were trained, and it was reported that the SVM algorithm achieved an accuracy rate of 88.7%. Gnanadass conducted a study to compare the effectiveness of various machine learning algorithms in predicting the risk of GDM [13]. The research utilized widely used machine learning models, including Random Forest, Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), and Logistic Regression. The results indicated that the Random Forest algorithm achieved the highest classification accuracy, reaching 92%. Ye et al. aimed to compare the performance of Random Forest, Gradient Boosting, Support Vector Machine, and traditional Logistic Regression methods for GDM prediction [41]. It was reported that the Random Forest method provided the best performance with an accuracy rate of 92.1%. Wei et al. recorded data using 67 indicators to predict GDM risk in early pregnancy. Popular machine learning algorithms such as Random Forest, Support Vector Machine (SVM), Gradient Boosting, and Logistic Regression were used [39]. They reported that the best results were obtained with the Random Forest algorithm, achieving a classification accuracy of 93.2%. Sumathi et al. proposed a machine learning-based ensemble classification model for the early diagnosis of GDM [34]. Missing values in the dataset were completed using appropriate methods, and the proposed model's performance was compared with traditional methods, achieving an ensemble model classification accuracy of 94.24%. In another study by Sumathi and Meganthan, a deep learning-based model for the early diagnosis of GDM was developed [35]. The proposed Deep Stacked Autoencoder model was used for GDM diagnosis, achieving a classification accuracy of 96.18%. Wang et al. aimed to demonstrate the applicability of ensemble learning methods for GDM prediction in clinical practice [38]. In this context, it was shown that methods such as XGBoost, Gradient Boosting, and Random Forest were more effective than using a single model. It was reported that XGBoost achieved the best performance with a classification accuracy of 94.7%.

In the study conducted by Kang et al. machine learning algorithms were used for GDM prediction in Asian women [18]. The modeling, conducted using Light Gradient Boosting Machine (LGBM) and XGBoost algorithms, showed improvements in AUC values ranging from 0.711 to 0.804. Kaya et al. collected data from 97 mothers, considering various factors such as maternal age, body mass index, gravida, parity, previous birth weight, smoking habits, first-visit venous plasma glucose levels, family history of diabetes, and oral glucose tolerance test results [19]. Their findings indicated that the eXtreme Gradient Boosting (XGB) classifier demonstrated the highest predictive performance, achieving an accuracy of 72.7%. Zhou et al. aimed to facilitate the use of machine learning models for step-by-step prediction of GDM and their integration into clinical decision-making processes [42]. In this context, they compared classifiers such as Random Forest, XGBoosting, Support Vector Machine (SVM), k-NN, and Logistic Regression. They reported that classification accuracies for prediction steps ranged from 76.6% to 92.2%.

This study aims to present a novel approach for the prediction of GDM using the data source realized in [34], [35]. In this method, two different approaches were followed. First, the original dataset was completed using KNN and Regression methods and then classified using classical KNN and SVM classifiers with the 10-fold cross-validation technique. In the second stage, both the original and the completed dataset were converted into grayscale images. These images were then augmented with a stride of 1 and classified separately using the deep learning technique, CNN, with training-test splits of 50%-50% and 70%-30%. The obtained results were compared with other studies conducted on the same dataset. After completing the missing data in the GDM dataset using KNN, converting it into grayscale images with a stride of 1, and classifying it with CNN using a 70%-30% train-test split, an accuracy rate of 97.91% was achieved. This result outperformed all previous studies conducted on the same dataset in the literature.

2. Methods

The data source used in this work taken from the following references [34], [35]. It consists of Gestational Diabetes Mellitus (GDM) records contains two classes (healthy and diabetic) and 3525 instances. It has a total of 3525 instances with the existence of 15 features. All samples have 15 features. These features with min and max values are: Age (20-45), No of Pregnancy (1-4), Gesstation in previous Pregnancy (0-2), BMI (13,3-45), HDL (15,70), Family History (0-1), Unexplained prenetal loss (0-1), Large Child or Birth Default (0-1), PCOS (0-1), Sys BP (90-185), Dia BP (60-124), OGTT (80-403), Hemoglobin (8,8-18), Sedentary Lifestyle (0-1), Prediabetes (0-1). Also, 2153 instances belong to class 0 and 1372 instances belong to class 1. Futhermore, there are missing values on BMI with 1081, HDL with 1001, Sys BP with 1705 and OGTT with 513. Table 1, represents the information of GDM dataset.

Table 1. Data Source Description

Names	Values
Number of Instances	3525
Number of Features	15
Number of Calss	2
Class-0 (healthy)	2153
Class-1 (diabetic)	1372
Missing of BMI	1081
Missing of HDL	1001
Missing of Sys BP	1075
Missing of OGTT	513

The flow diagram of the proposed method is presented step by step in Fig. 1. As seen in the figure, the data source is analyzed using two different approaches. In the first approach, the original dataset is classified using well-known machine learning techniques such as KNN and SVM with the k-fold technique, and the corresponding performance metrics are provided. On the other hand, the original data is transformed into grayscale images without altering the feature dimension (rows). In this transformation process, the original data is first segmented into 15×15 patches to generate primary images. To increase the number of images, a stride size of 1 (stride=1) is applied, shifting the window forward by one row at a time to generate multiplexed images. The primary and multiplexed images are then classified using a CNN-based deep learning model, and performance

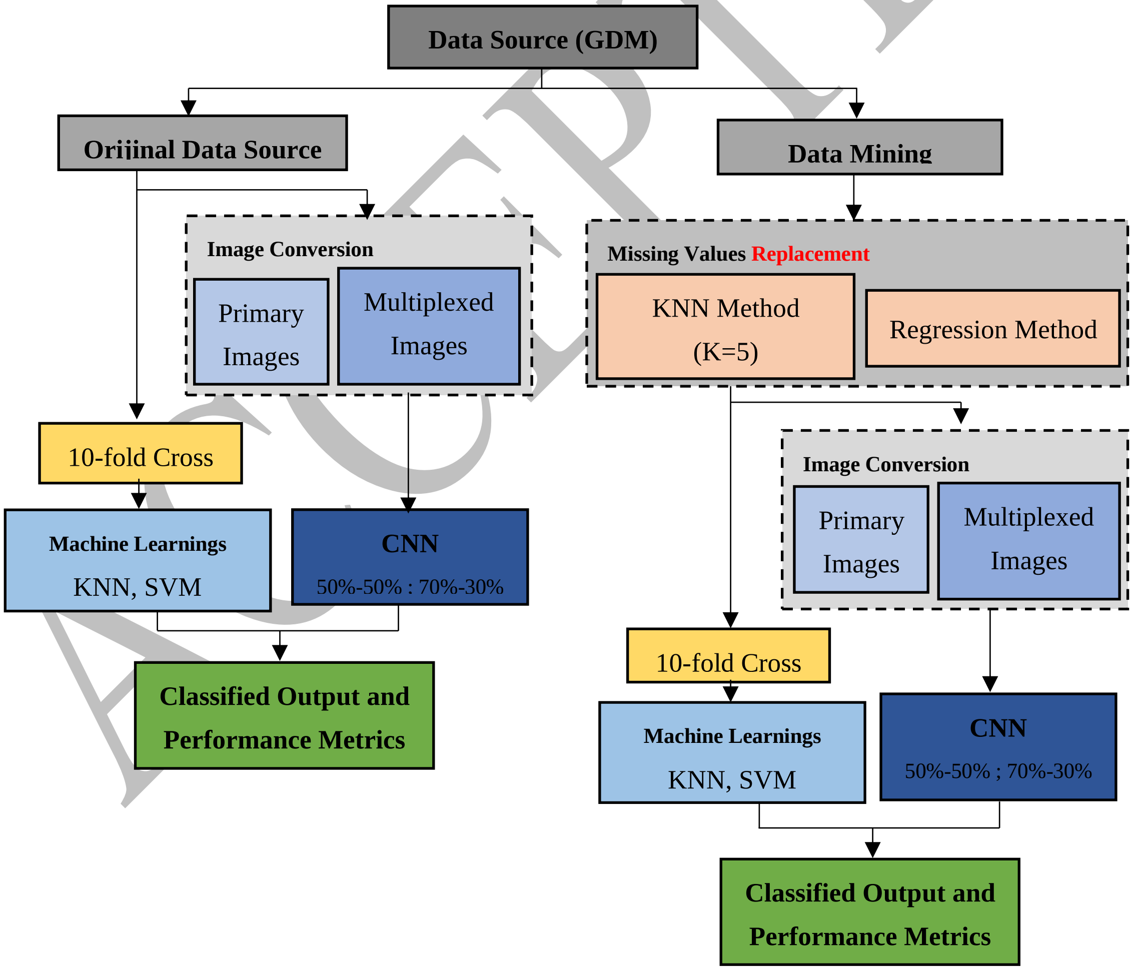


Fig. 1. The Flow Chart of Proposed Model

metrics are obtained. In the second approach, missing values in the original dataset are filled using data mining techniques such as KNN and Regression. After completing the missing data, the same procedures applied in the first approach are repeated, and the final results are obtained.

The missing data problem is a common issue encountered in many datasets. Properly handling missing data can directly impact the success of data analysis and machine learning models. The most effective and widely used methods for the filling of missing data are KNN and Regression methods [27], [29].

The K-Nearest Neighbors (KNN) method is an effective and widely used statistical approach for handling missing data [6]. In this method, an instance with missing values is compared with its most similar neighbors, and the missing data is estimated and filled accordingly [36]. The KNN-based missing data imputation method predicts missing values in a dataset by leveraging the similarity between data points. Compared to simple imputation techniques such as mean or median imputation, KNN provides more accurate results since it preserves the natural structure of the data [17]. The KNN method is based on estimating missing values by comparing each instance with missing data to its nearest neighbors. By assigning weights to neighbors based on their distances, a more accurate estimation can be achieved, ensuring that closer neighbors have a greater influence. Weighted averaging improves predictions by considering the similarities between data points. The contribution of nearest neighbors is increased using equation (1) below, leading to a more precise estimation:

$$x_{missing} = \frac{\sum_{i=1}^{K} \frac{x_{neighbor_i}}{d\left(x_{missing}, x_{neighbor_i}\right)}}{\sum_{i=1}^{K} \frac{1}{d\left(x_{missing}, x_{neighbor_i}\right)}} \qquad (1)$$

here; $x_{missing}$ represents the missing value to be imputed, $x_{neighbor_i}$ denotes the known value of the ith neighbor, $d\left(x_{missing}, x_{neighbor_i}\right)$ represents the Euclidean distance and K is the number of nearest neighbors considered. In this study, missing values were imputed using the weighted values of the five nearest neighbors.

Another powerful method used for missing data imputation is the Regression method. Linear regression is widely used for predicting missing values by modeling a linear relationship between independent and dependent variables. In this approach, missing values in the dataset are treated as dependent variables, while the other observed features are considered as independent variables for prediction. Linear regression is the most commonly used regression model, as it establishes a linear relationship between independent and dependent variables [2], [23]. The primary goal of linear regression is to explore the relationships between the given independent variables and the dependent variable and to estimate missing values based on these relationships. The fundamental function of linear regression is to predict the dependent variable (missing value) as a linear combination of independent variables [12]. Mathematically, it is expressed as follows:

$$y_{missing} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \in \tag{2}$$

here; $y_{missing}$ represents the missing value to be imputed, β_0 is the constant term, $\beta_1, \beta_2, \ldots, \beta_n$ are the regression coefficients, $x_1, x_2, \ldots, x_n$ are the independent variables, and $\in$ is the error term.

The data source consists of 2153 instances of Class-0, 1372 instances of Class-1, and 15 determining features. Fig.2 summarizes how the primary image and multiplexed image extraction processes are carried out, using Class-0 samples to avoid confusion. Initially, the original Class-0 data source, which contains missing values, is divided into 15×15 matrix patches. As shown by the black arrows in Figure-2, each patch is then converted into a grayscale image to form an image. As a result of this step, 143 primary Class-0 images (15×15) and 91 Class-1 images are created using the same logic. Subsequently, to increase the number of images, the stride size of 1 (stride=1) is chosen, and the window starts shifting 1 row down each time. New 15×15 grayscale multiplexed images are generated, as shown by the red arrows in Figure 1. For Class-0, 2139 multiplexed images of size 15×15 are created, while for Class-1, the number of images is 1358. In the next phase, after filling the missing data in the dataset using KNN and Regression, the primary images and multiplexed images are recreated in the same manner [4], [14].

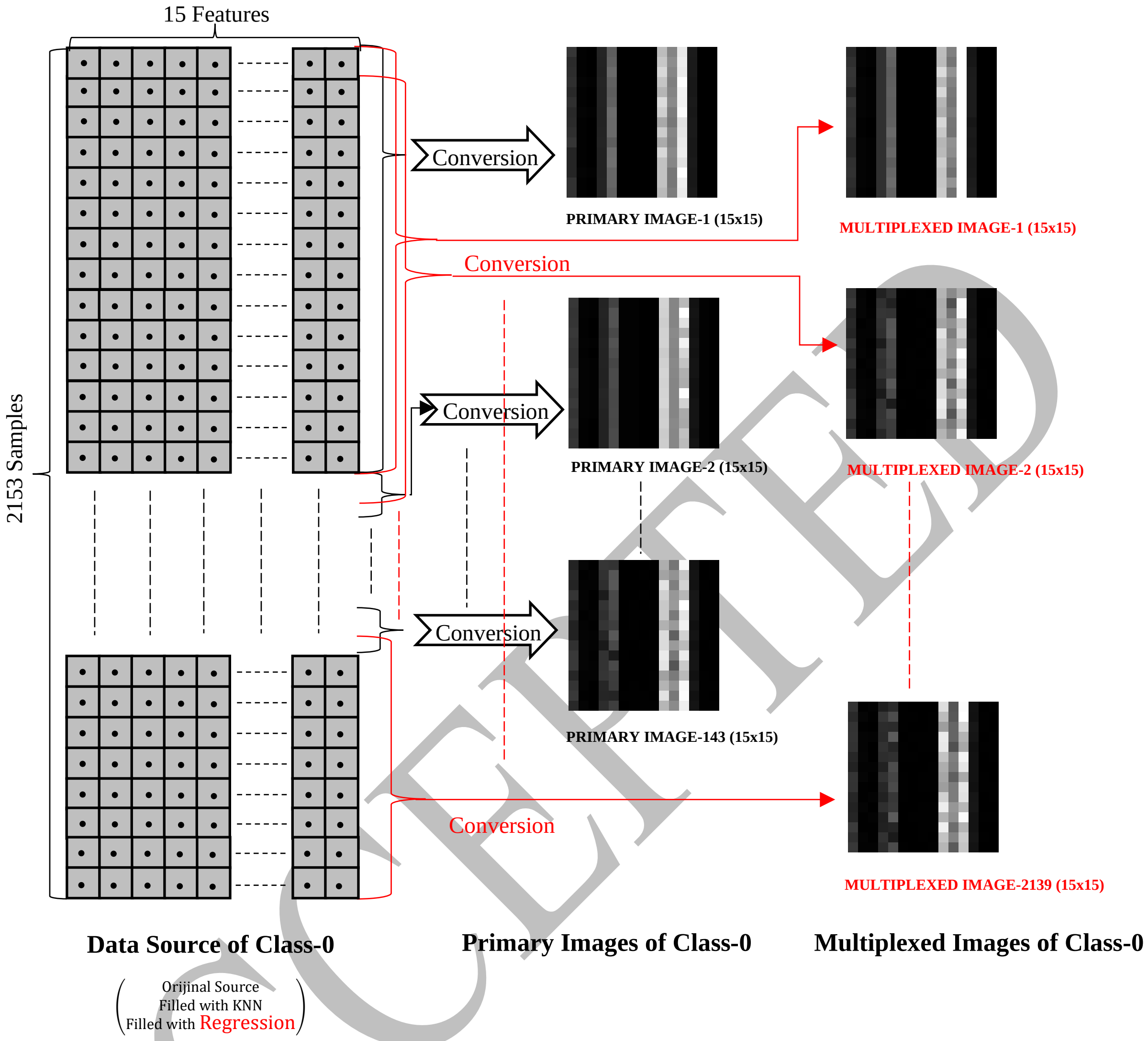


Fig. 2. Image Conversation Process of Data Source on Class-0

Powerful methods widely used in classification and pattern recognition tasks, such as KNN, SVM, and CNN, have been applied in this study. To enhance the reliability of the KNN and SVM classifier's accuracy, the 10-fold cross-validation technique has been utilized. In the k-fold cross-validation technique, the dataset is divided into multiple subsets (folds). In each iteration, one fold is set aside as the test set, while the remaining folds are used to train the model. This process is repeated k times, ensuring that each subset serves as the test set once, leading to a more reliable evaluation of the model's performance. This process is repeated k times, ensuring that each subset is used as the test data exactly once. As a result, the overall performance of the model is calculated by averaging the evaluation metrics obtained from all iterations [5]. The mathematical formulas for Accuracy (GDM) are shown in equations (3), (4), and (5).,

$$\mathrm{Accuracy(TS)} = \frac{\sum_{\mathrm{i=1}}^{|\mathrm{TS}|} \mathrm{estimate(n_i)}}{|\mathrm{TS}|}; \quad n_i \in TS \tag{3}$$

$$\mathrm{estimate(n)} = \begin{cases} 1 & \text{if estimate(n) = cn} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$Accuracy(GDM) = \frac{\sum_{i=1}^{|k|} accuracy(TS_i)}{|k|} \tag{5}$$

here, TS is the test set (fold) to be classified, nϵTS, cn is the class of the n and estimate(n) is the classification result of n estimated by networks.

For deep learning, the CNN model was tested by splitting the data into 50%-50% and 70%-30% training-test parts, and the performance of the model on image classification was compared. Additionally, performance evaluation metrics are crucial to demonstrate the success and reliability of the classifiers [36]. These parameters are given in equations (6-9).

$$\text{Recall (Sensitivity)} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{6}$$

$$\text{Specificity} = \frac{\mathrm{TN}}{\mathrm{TN+FP}} \tag{7}$$

$$\text{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{8}$$

$$\mathrm{F1-Score} = 2\mathrm{x}\frac{\text{Precision x Recall}}{\text{Precision+ Recall}} \tag{9}$$

where, TP is True Positive, TN is True Negative, FP is False Positive, FN is False Negative.

Recall is the ratio of true positives to the number of positive instances correctly identified by the model. Precision shows how accurate the positive predictions made by the model are. Specificity indicates how successful the model is at identifying negative classes. F1-Score is the harmonic mean of Precision and Recall, taking into account both the accuracy and sensitivity of the model.

The K-Nearest Neighbors (KNN) algorithm is widely used in machine learning due to its simplicity and effectiveness. KNN determines the class of a new data point by considering the class labels of its nearest neighbors, thus performing the classification task [10], [28]. The basic principle of the KNN algorithm is to determine the K nearest neighbors of a data point and predict the class of the new data point based on the class labels of these neighbors. In this process, distances between data points are typically calculated using Euclidean distance, as shown in equation 10 [30].

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right)^2} \quad (10)$$

here; xi and xj represent the two data points being compared, xik and xjk denote the k-th feature values of the respective points, and n is the number of features.

Support Vector Machine (SVM) is fundamentally based on the principle of finding a hyperplane that best separates the data [9]. The primary goal is to separate different classes with a hyperplane that has the widest marginal gap. This hyperplane works by maximizing the margin between different classes. The margin refers to the distance between the hyperplane and the nearest data points, known as support vectors, which are the critical points that influence the position of the hyperplane [7]. The SVM classification method is mathematically expressed in equations (11-14).

The training data (xi) belonging to two separated classes (yi),

$$\{x_i, y_i\}, \quad i = 1,2, \dots, N, \quad y_i \in \{-1, +1\}, \quad x_i \in R^n \, ; \quad (11)$$

represented with the optimal hyperplane,

$$(w.x_i) + b = 0 \, ; \quad (12)$$

Optimal hyperplane with the largest margin can be formulated as follows:

$$\begin{aligned} x_i.w + b &\geq +1 \quad for\ y_i = +1 \\ x_i.w + b &\leq +1 \quad for\ y_i = -1 \end{aligned} \quad (13)$$

which is equivalent to

$$\begin{cases} w^T\varphi(x_i) + b \geq +1, & if\ y_i = +1 \\ w^T\varphi(x_i) + b \leq -1, & if\ y_i = -1 \end{cases} \quad \rightarrow \quad y_i[w^T\varphi(x_i) + b] \geq 1 \quad (14)$$

where $\varphi(:)$ is a function which maps the input space into a higher dimensional spice.

Convolutional Neural Network (CNN) is a key approach in deep learning, particularly distinguished by their success in visual recognition and classification tasks. Inspired by the visual cortex of biological systems, CNNs can automatically and adaptively learn spatial features from data in a hierarchical manner [15]. Unlike traditional machine learning methods, CNNs effectively learn by capturing meaningful patterns from the data. The CNN architecture generally includes convolutional layers, pooling layers, and fully connected layers, each serving a distinct purpose in processing and classifying data [4], [20]. The convolutional layer performs feature extraction by sliding a learnable filter over the input data matrix. Mathematically, the convolution operation between a 2D input matrix and a filter is expressed in Equation (15).

$$S(i,j) = (X * F)(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(i+m, j+n).F(m,n) \quad (15)$$

here; S(i,j): is the pixel value at position (i,j) of the feature map, X: is the input image, F: s the convolution filter, M and N: are the filter dimensions, * : denotes the convolution operation.

Convolutional layers, as a result of this operation, detect features such as edges, textures, and shapes [4], [14]. After the convolution operation, a nonlinear activation function is applied. The most commonly used activation function is ReLU (Rectified Linear Unit), which is defined in (16).

$$f(x) = \max(0, x) \quad (16)$$

This function sets negative values to zero and leaves positive values unchanged. The activation function allows the network to learn nonlinear relationships. The pooling layer is used to reduce the size of the feature maps and make the model more computationally efficient, thereby reducing the computational load. The most common method is max-pooling, which is expressed as in (17).

$$P(i,j,k) = \underbrace{max}_{(m,n)\in R} S(i+m, j+n, k) \quad (17)$$

here; P(i,j,k): is the (i,j) value in the pooled feature map, R: is a specific pooling window and S(i+m,j+k,k): is the feature map after convolution.

The pooling operation allows the model to gain local invariance. The 2D feature maps obtained from the pooling layer are flattened before being passed to the fully connected layer. In the fully connected layer, each neuron is connected to all input features. It performs the classification task by converting the learned features into class probabilities [22], [33], [40]. Mathematically, this is expressed as in equation (18);

$$y_j = f\left(\sum_{i=1}^{N} w_{ij}.x_i + b_j\right) \quad (18)$$

here; yj is the output of the j-th neuron, xi is the i-th input feature, wij is the weight of the connection from i to j, bj is the bias value and f(.) is the activation functio.

In the output layer, for classification problems, the softmax function is used to calculate probabilities for each class with equation (19).

$$\widehat{y_i} = \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)} \tag{19}$$

here; y_i: is the predicted probability for the i-th class, zi: is the activation of the i-th neuron in the fully connected layer, C: is the number of classes.

The softmax function normalizes the probability of each class between 0 and 1 and adjusts them such that their sum equals 1.

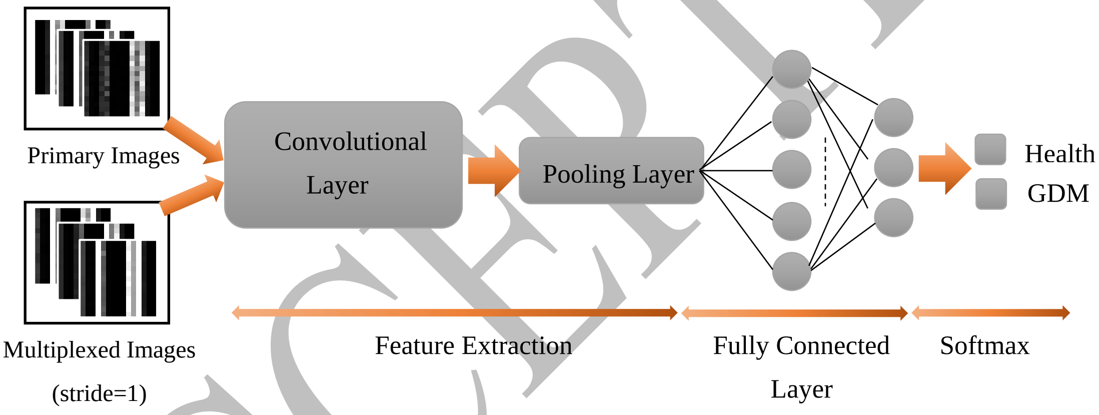


Fig. 3. The Main Concept of CNN

As shown in Fig. 3, the CNN architecture used in the study includes one convolution layer with a kernel size of 8, with the number of kernels ranging from 5 to 20. Additionally, the maximum pooling layer has a kernel size of 2, and the stride value was chosen as 2. Furthermore, the network structure was tested with both 50%-50% and 70%-30% training-test splits.

3. Results

The performance metrics obtained from classifying the raw GDM dataset and the dataset completed using KNN and Regression methods with machine learning algorithms are presented in the Table 2.

KNN and SVM classifiers executed by using the 10-fold cross-validation technique on both the raw GDM data source and the dataset where missing values have been filled using KNN and Regression methods. The results highlight both the effectiveness of the classifiers and the impact of missing data imputation methods on classification performance.

Table 2. Classification Results of Machine Learning Methods

METHOD		**Classification Results (%) with k=10 fold**				
		Recall	**Specificity**	**Precision**	**F1-Score**	**Accuracy**
Orijinal Data Source	KNN	30,33	97,92	89,89	45,22	71,70
	SVM	33,67	96,03	83,93	48,01	71,93
Missing Values filled with KNN	KNN (5-NN)	97,91	95,98	97,4	97,65	97,13
	SVM	96,84	97,45	98,36	97,59	97,08
Missing Values filled with Regression	KNN (5-NN)	98,46	95,89	97,31	97,87	97,42
	SVM	96,98	97,89	98,6	97,8	97,33

When using the original dataset containing the raw GDM data source values, the KNN method achieved an accuracy of 71.70%, while the SVM method showed an accuracy of 71.93%. However, the low Recall values (KNN: 30.33%, SVM: 33.67%) indicate that the methods struggled to identify the patient class. On the other hand, the high Specificity values (KNN: 97.92%, SVM: 96.03%) demonstrate that healthy individuals were classified correctly. These results show that the raw data, due to missing values, suffered from information loss and its direct use negatively impacted diagnostic performance.

Imputing missing values using the KNN method led to a noticeable improvement in classification performance. The KNN algorithm achieved an accuracy of 97.13%, while the SVM algorithm achieved an accuracy of 97.08%. There was also a significant increase in Recall, Precision, and F1-score values. For KNN, a Recall of 97.91% and Precision of 97.4% demonstrated that the classifier was able to accurately detect both patients and healthy individuals. Similarly, imputing missing values using the Regression method further improved classification performance. The highest classification accuracy achieved with the KNN method was 97.42%, while the accuracy with SVM was quite close, at 97.33%. In particular, the KNN algorithm's Recall value of 98.46% indicates that patient individuals were identified with the highest accuracy. The SVM algorithm also achieved remarkable results with an F1-score of 97.8%.

Table 3. CNN Classification Results

CNN		**Classification Results (%)**											
		50%-50%						**70%-30%**					
		Rc.	**Spc.**	**Pr.**	**F1**	**Class Acc..**	**Acc.**	**Rc.**	**Spc.**	**Pr.**	**F1**	**Class Acc.**	**Acc.**
Primary Images	Cls-0	100	86,67	92,21	95,95	94,83	94,83	100	88,89	93,48	96,63	95,71	95,71
	Cls-1	86,67	100	100	92,86	94,83		88,89	100	100	94,11	95,71	
Multiplexed Images (stride=1)	Cls-0	96,63	98,23	98,85	97,73	97,25	**97,25**	97,35	97,3	98,27	97,81	97,33	97,33
	Cls-1	98,23	96,63	94,88	96,53	97,25		97,3	97,35	95,88	96,59	97,33	
Primary Images filled with KNN	Cls-0	100	86,67	92,21	95,95	94,83	94,83	97,67	88,89	93,33	95,46	94,29	94,29
	Cls-1	86,67	100	100	92,86	94,83		88,89	97,67	96	92,31	94,29	
Multiplexed Images filled with KNN (stride=1)	Cls-0	98,88	92,49	95,4	97,11	96,4	96,4	98,91	96,31	97,69	98,3	97,91	**97,91**
	Cls-1	92,49	98,88	98,13	95,22	96,4		96,31	98,91	98,25	97,27	97,91	
Primary Images filled with Regression	Cls-0	100	86,67	92,21	95,95	94,83	94,83	100	88,89	93,48	96,63	95,71	95,71
	Cls-1	86,67	100	100	92,86	94,83		88,89	100	100	94,12	95,71	
Multiplexed Images filled with Regression (stride=1)	Cls-0	98,97	94,55	96,62	97,78	97,25	**97,25**	97,82	97,05	98,13	97,97	97,52	97,52
	Cls-1	94,55	98,97	98,32	96,4	97,25		97,05	97,82	96,58	96,81	97,52	

Pr.: Precision, **Rc.:** Recall, **Spc.:** Specificity, **F1:** F1-Score, **Class Acc.:** Class Accuracy, **Acc.:** Accuracy

Table 3 shows the performance metrics obtained from classifying grayscale images generated with the GDM data source and after missing values in this dataset were filled using the KNN and regression methods, through a CNN structure. The classification is performed on the following: primary images generated with the GDM data source, multiplexed images obtained by choosing a stride of 1, primary images obtained after missing values are filled using the KNN method, multiplexed images generated with stride=1, and primary images generated after filling missing values using the Regression method, followed by multiplexed images with stride=1. The classification is done for each case with a 50%-50% and 70%-30% (test-train) split.,

For the classification with grayscale images generated from the GDM data source, the overall accuracy of the "Primary Images" dataset with the 50%-50% split ratio was found to be 94.83%, with recall and precision values for Class-0 being 100% and 92.21%, respectively, and for Class-1, recall was 86.67% and precision was 100%. This indicates that the model's positive predictions are largely accurate. The overall accuracy for both split ratios was 94.83% and 95.71%, respectively. Similarly, for the "Multiplexed Images" dataset with the same split ratio, the overall accuracy was 97.25%, with recall values of 96.63% for Class-0 and 98.23% for Class-1, showing a more balanced performance. With the 70%-30% split ratio, the "Primary Images" dataset provided an overall accuracy of 95.71%, while the "Multiplexed Images" dataset achieved an overall accuracy of 97.33%.

After filling missing values with the Regression method, high accuracy values were similarly obtained, with images generated with a stride of 1 and the 70%-30% split ratio achieving an overall accuracy of 97.52%. After filling missing values using the KNN method, high accuracy values were also obtained, especially for the "Multiplexed Images" dataset with stride=1, where the 70%-30% split ratio resulted in an overall accuracy of 97.91%. Notably, recall (98.91% for Class-0, 96.31% for Class-1), precision (97.69% for Class-0, 98.25% for Class-1), and F1-Score (98.3% for Class-0, 97.27% for Class-1) values demonstrate superior performance in GDM diagnosis. This result shows that the KNN method outperforms other classification algorithms and data processing methods.

4. Discussion

When reviewing the comparison table in Fig. 4, it is observed that the original dataset significantly lowered the performance of the classification algorithms (KNN: 71.7%, SVM: 71.93%). This is due to missing values in the original data, which negatively affect the classifier performance. However, filling missing data significantly improves the classification performance, and it is observed that the method of filling missing values with Regression performs slightly better than the

KNN method. In terms of classification results, there was no significant difference between KNN (97.42%) and SVM (97.33%). Both models achieved very high accuracy after missing values were filled. This indicates that a balanced data distribution positively affects classification performance.

When the original dataset containing the raw GDM data source values is used, the KNN method provides an accuracy of 71.70%, while the SVM method shows an accuracy of 71.93%. However, the low recall (KNN: 30.33%, SVM: 33.67%) values indicate insufficient recognition of the patient class. High specificity values (KNN: 97.92%, SVM: 96.03%) indicate that healthy individuals are classified correctly. However, these results show that the raw data suffers from information loss due to missing values, and its direct use negatively affects diagnostic performance.

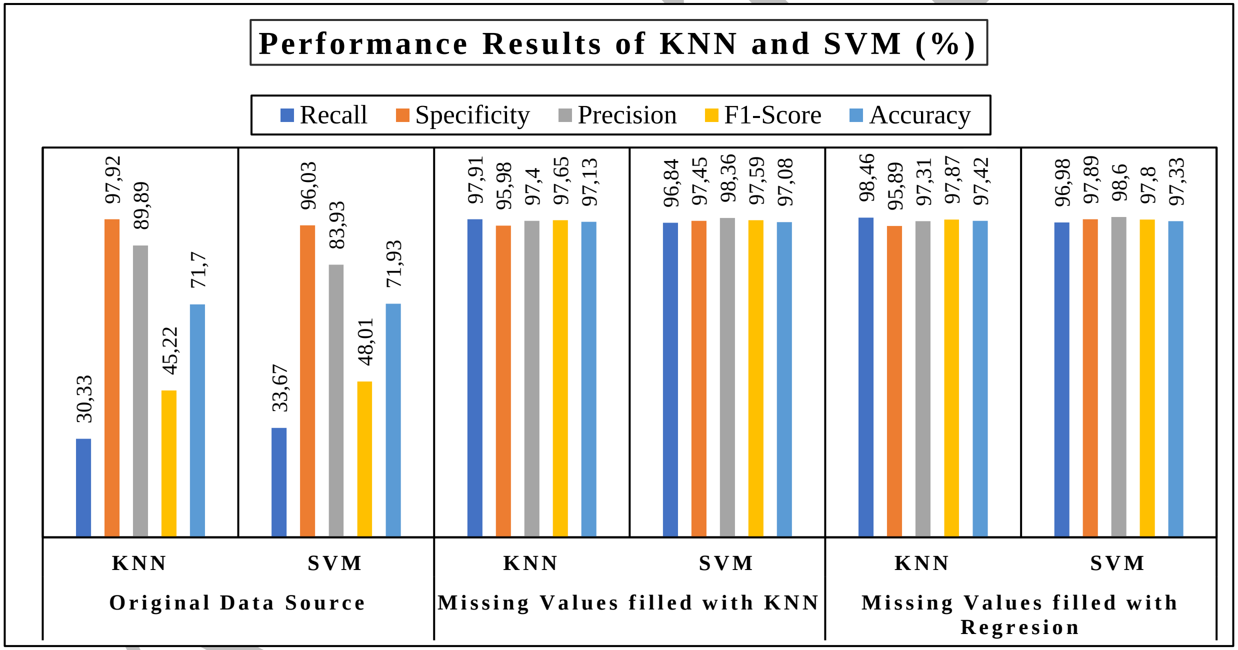
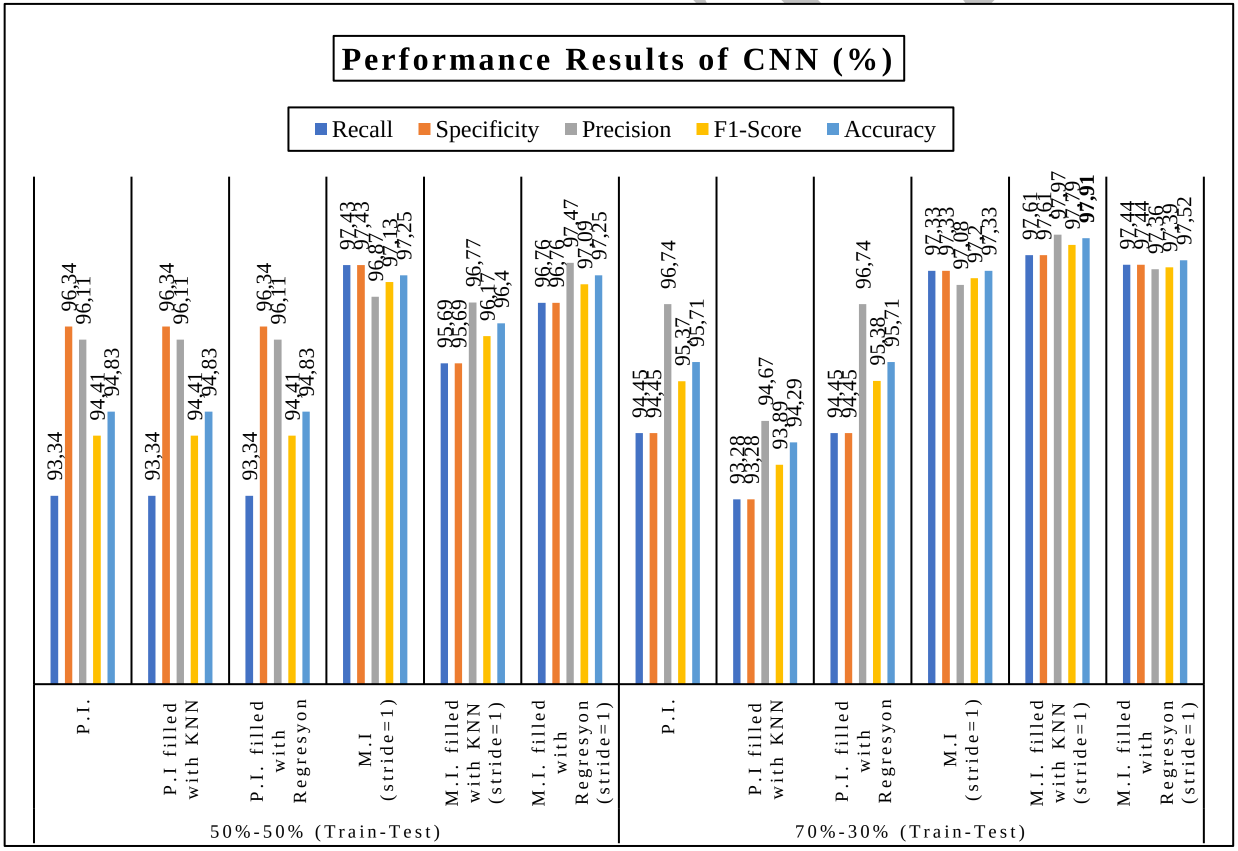


Fig. 4. Performance Results of KNN and SVM Methods on GDM Data Source

Fig. 5 presents the performance results of the CNN, showing the CNN performance metrics for both primary and multiplexed images generated from the GDM data source, as well as for the primary and multiplexed images generated after filling the missing values of the GDM data source using the KNN and Regression methods (50%-50% and 70%-30% splits). All values, except for accuracy, represent the average of the results obtained for Class-0 and Class-1.

When the comparison table in Fig. 5 is examined, it is immediately noticeable that, unlike the performance of KNN and SVM methods, the CNN method provides higher performance values when classification is performed on images derived from the original data (missing values) set (P.I.). This can be attributed to the unique feature extraction technique and strength of the CNN architecture. The results obtained using the 70%-30% train-test split are more successful and demonstrate more balanced performance compared to those obtained with the 50%-50% split across all methods in the CNN architecture. The best result was achieved on the dataset completed with KNN, where the accuracy of the multiplexed grayscale images with stride=1 reached 97.91%. This value is supported by recall and specificity values of 97.61%, as well as precision and F1-Score values of 97.79%, showing that both the patient and healthy mothers were classified correctly with high accuracy.



P.I.: Primary Images, **M.I.:** Multiplexed Images

Fig. 5. Performance Results of CNN Method on GDM Data Source

At the same time, it is evident that the regression results for the multiplexed dataset after data mining are also quite high. It would not be wrong to say that both methods can be preferred.

As a result, the classification results obtained from both the primary and multiplexed images show that multiplexed images offer a more balanced performance. These results indicate that multiplexed images provide a more reliable dataset for classification. Additionally, the classification performances of classical methods have significantly improved after the data completion processes. These methods are now deemed more reliable for classification. When both Fig. 5 and Fig. 6 are evaluated together, the obtained performance results highlight the importance of missing data processing techniques in GDM diagnosis and emphasize the effectiveness of machine learning models.

This study demonstrates a significant improvement compared to previous methods reported in the literature as seen on Table 4. In this work, we proposed a hybrid approach combining Data Mining techniques with CNN for classification tasks, achieving a classification accuracy of 97.91%.

Table 4. The Comparison of Classification Accuracies on GDM Data Source

Study	**Method**	**Classification Accuracy (%)**
[34]	Ensemble Model	94,24
[35]	OD-DSAE	96,18
This Paper	Data Mining + CNN	**97,91**

When compared to the study [34], which utilized an Ensemble Model and achieved an accuracy of 94.24%, our method outperforms by 3.67 percentage points. The superior performance of our approach can be attributed to the integration of data mining techniques, which effectively preprocess, extract meaningful features from the dataset and image conversion process combined with the powerful feature learning capabilities of CNNs. Similarly, our method also outperforms the work [35], which uses the OD-DSAE (Deep Stacked Autocoder for Outlier Detection) model and achieves 96.18% accuracy. While OD-DSAE is effective in handling outliers and learning hierarchical representations, our hybrid approach of data mining and CNN achieves a 1.73% point improvement in accuracy by using both structured feature extraction, image conversion, and deep learning. This highlights the importance of combining traditional data mining techniques and image conversion with modern deep learning architectures. In conclusion, our results demonstrate that the combination of Data Mining and CNN offers a robust and effective solution for classification tasks, outperforming existing methods in terms of accuracy. At the same time, the high performance values of combining traditional data mining techniques with KNN and SVM should also be taken into consideration.

5. Conclusion

In this study, a novel approach was presented to predict the fastest and most effective treatment method for GDM and it was aimed to improve the results of previous studies conducted for the same purpose. Accordingly, the GDM data source was classified using machine learning algorithms such as KNN and SVM both in its original form and after being processed with missing values imputed using KNN and Regression methods. Additionally, the original and the imputed dataset were converted to gray-scale images and both datasets were multiplied by selecting stride=1 and classified individually with the CNN structure. When the obtained performance values compared, the best result was achieved using the CNN model (train: 70%, test: 30%) with 97.91% accuracy, based on images augmented with stride = 1 after adding missing data using the KNN method. At the same time, the classification results of machine learning algorithms such as KNN and SVM with the support of 10-fold-cross validation technique after filling the missing data with the data processing methods also provided remarkable results. The results confirm that this study achieves the highest classification accuracy reported in the literature on the given GDM dataset. It can be clearly stated that applying a data mining method to impute missing values, followed by converting the dataset into images based on certain criteria and classifying it with CNN, is the most effective approach for predicting GDM. Therefore, this hybrid approach of data mining and CNN method could provide physicians with a more reliable solution for GDM assessment. Additionally, this approach has been applied for the first time in the literature and can serve as an initial decision-support system for physicians before resorting to other medical diagnostic methods.

Reference

[1] Afsaneh E., Sharifdini A., Ghazzaghi H., Ghobadi M.Z., *Recent applications of machine learning and deep learning models in the prediction, diagnosis, and management of diabetes: a comprehensive review,* Diabetology & Metabolic Syndrome, 2022, 14(196), DOI: 10.1186/s13098-022-00969-9.

[2] Alruhaymi A.Z. and Kim C.J., *Study on the Missing Data Mechanisms and Imputation Methods,* Open Journal of Statistics, 2021,11(4):477-492, DOI: 10.4236/ojs.2021.114030.

[3] Amarnath, S., Selvamani, M., & Varadarajan, V., *Prognosis Model for Gestational Diabetes Using Machine Learning Techniques.* Sensros and Metarials, 2021,3(9):3011-3025, DOI:10.18494/SAM.2021.3119.

[4] Bascil M.S., *Convolutional Neural Network to Extract the Best Treatment Way of Warts Based on Data Mining,* Revue d'Intelligence Artificielle, 2019,33(3):165-170, DOI: 10.18280/ria.330301.

[5] Bascil M.S., *A New Approach on HCI Extracting Conscious Jaw Movements Based on EEG Signals Using Machine Learnings,* Journal of Med. Syst., 2018,42:169, DOI: 10.1007/s10916-018-1027-1.

[6] Batista G.E., and Monard M.C., *An analysis of four missing data treatment methods for supervised learning,* Applied Artificial Intelligence, 2003,17(5-6):519–533, DOI: 10.1080/713827181.

[7] Ben-Hur A. and Weston J., *A user's guide to support vector machines,* Methods in Molecular Biology, 2009,609:223-239, DOI: 10.1007/978-1-60327-241-4_13.

[8] Benham J.L., Gingras V., McLennan N.M., Most J., Yamamoto J.M., Aiken C.E,. et al., *Precision gestational diabetes treatment: a systematic review and meta-analyses.* Communications Medicine, 2023, 3(1):135, DOI: 10.1038/s43856-023-00371-0.

[9] Cortes C. and Vapnik V., *Support-vector networks,* Machine Learning, 1995,20(3):273-297, DOI:10.1007/BF00994018.

[10] Cover T. and Hart P., *Nearest neighbor pattern classification,* IEEE Transactions on Information Theory, 1967,13(1):21-27, DOI: 10.1109/TIT.1967.1053964.

[11] DeFronzo R.A., Ferrannini E., Groop L., Henry R.R., Herman W.H., Holst J.J., et al., *Type 2 diabetes mellitus,* Nature Reviews Disease Primers, 2015, 1(15019), DOI: 10.1038/nrdp.2015.19.

[12] Emmanuel T., Maupong T., Mpoeleng D., Semong T., Mphago D., Tabona O., *A survey on missing data in machine learning,* Journal of Big Data 2021,8:140, DOI: 10.1186/s40537-021-00516-9.

[13] Gnanadass I., *Prediction of Gestational Diabetes by Machine Learning Algorithms*, IEEE Potentials, 2020,39(6):32-37, DOI: 10.1109/MPOT.2020.3015190.

[14] Gorur K., Bozkurt M.R., Bascil M.S., Temurtas F., *GKP signal processing using deep CNN and SVM for tongue-machine interface,* Traitement du Signal, 2019;,6(4):319-329, DOI: 10.18280/ts.360404.

[15] Görür K., Bozkurt M.R., Bascil M.S., Temurtas F., *Tongue-Operated Biosignal over EEG and Processing with Decision Tree and kNN,* Academic Platform-Journal of Engineering Science, 2021,9(1):112-125, DOI: 10.21541/apjes.583049.

[16] Huang Y., McCullagh P., Black N., Harper R., *Feature selection and classification model construction on type 2 diabetic patient's data*, Artif. Intell. Med., 2007,41(3):251-262, DOI: 10.1016/j.artmed.2007.07.002.

[17] Junninen H., Niska H., Tuppurainen K., Ruuskanen J., Kolehmainen M., *Methods for imputation of missing values in air quality data sets,* Atmospheric Environment, 2004,38(18):2895-2907, DOI: 10.1016/j.atmosenv.2004.02.02.

[18] Kang B.S., Lee S.U., Hong S., Choi S.K., Shin J.E., et al., *Prediction of gestational diabetes mellitus in Asian women using machine learning algorithms,* Scientific Reports, 2023,13(13356), DOI:10.1038/s41598-023-39680-8.

[19] Kaya Y., Bütün Z., Çelik Ö., Salik E.A., Tahta T., Yavuz A.A., *The early prediction of gestational diabetes mellitus by machine learning models,* BMC Pregnancy and Childbirth, 2024,24(1):574, DOI:10.1186/s12884-024-06783-7.

[20] LeCun Y., Bottou L., Bengio Y., Haffner P., *Gradient-based learning applied to document recognition,* Proceedings of the IEEE, 1998,86(11):2278-2324, DOI: 10.1109/5.726791.

[21] Liao, H., Zhang, X., Zhao, C., Chen, Y., Zeng, X., & Li, H., *LightGBM: an efficient and accurate method for predicting pregnancy diseases.* Journal of Obstetrics and Gynaecology, 2021,42(4):620-629.

[22] Litjens G., Kooi T., Bejnordi B.E., Setio A.A., Ciompi F., et al., *A survey on deep learning in medical image analysis,* Medical Image Analysis, 2017,42:60-88, DOI: 10.1016/j.media.2017.07.00.

[23] Little R.J.A.. and Rubin D.B., *Statistical Analysis with Missing Data*, Wiley, 2019.

[24] Magoulas, G.D., Prentza A., *Machine Learning in Medical Applications,* Springer, Berlin, 2001. DOI: 10.1007/3-540-44673-7_19.

[25] McIntyre H.D., Catalano P., Zhang C., Desoye G., Mathiesen E.R., Damm P., *Gestational diabetes mellitus.* Natura Reviews Disease Primers, 2019, 5(1):47, DOI: 10.1038/s41572-019-0098-8.

[26] Ozer I., Karaca A.C., Ozer C.K., Gorur K., Kocak I., Cetin O., *The exploration of the transfer learning technique for* Globotruncanita *genus against the limited low-cost light microscope images,* Signal, Image and Video Proesssing, 2024,18:6363-6377, DOI: 10.1007/s11760-024-03322-x.

[27] Papailiou I., Spyropoulos F., Trichakis I., Karatzas, G.P., *Artificial Neural Networks and Multiple Linear Regression for Filling in Missing Daily Rainfall Data,* Water, 2022,14(18):2892, DOI:10.3390/w14182892.

[28] Peterson L.E. *K-Nearest Neighbor,* Scholarpedia, 2009,4(2):1883, DOI: 10.4249/scholarpedia.1883.

[29] Qi X., Guo H., Wang W., *A reliable KNN filling approach for incomplete interval-valued data,* Engineering Applications of Artificial Intelligence, 2021,100:104175, DOI: 10.1016/j.engappai.2021.104175.

[30] Rana M. and Bhushan M., *Machine learning and deep learning approach for medical image analysis: diagnosis to detection,* Multimedia Tools Applications, 2023,82:26731-26769, DOI: 10.1007/s11042-022-14305-w.

[31] Sarwar M.A., Kamal N., Hamid W., Shah M.A., *Prediction of Diabetes Using Machine Learning Algorithms in Healthcare. 24th International Conference on Automation and Computing (ICAC) 2018*, pp. 1-6, Newcastle Upon Tyne, UK, DOI: 10.23919/IConAC.2018.8748992.

[32] Shen J., Chen J., Zheng Z., Zheng J., Liu Z., Song J., et al., *An Innovative Artificial Intelligence-Based App for the Diagnosis of Gestational Diabetes Mellitus (GDM-AI): Development Study*, Journal of Medical Internet Research, 2020,22(9):e21573, DOI: 10.2196/21573.

[33] Shen D., Wu G., Suk H.I., *Deep learning in medical image analysis*, Annual Review of Biomedical Engineering, 2017,19:221-248, DOI: 10.1146/annurev-bioeng-071516-044442.

[34] Sumathi A., Meganathan S., Ravisankar S.V., *An intelligent gestational diabetes diagnosis model using deep stacked autoencoder,* Computers, Materials & Continua, 2021,69(3):3109-3126, DOI:10.32604/cmc.2021.017612.

[35] Sumathi A., Meganathan S., *Ensemble Classifier Technique to Predict Gestational Diabetes Mellitus (GDM),* Computer Systems Science and Engineering, 2022,40(1):313-325, DOI:10.32604/csse.2022.017484.

[36] Troyanskaya O., Cantor M., Sherlock G., Brown P., Hastie T., et al., *Missing value estimation methods for DNA microarrays*, Bioinformatics 2001,17(6):520-525, DOI: 10.1093/bioinformatics/17.6.520.

[37] Trujillo A.L., *Insulin Analogs and Pregnancy*, Diabetes Spectrum, 2007, 20 (2):94-101, DOI: 10.2337/diaspect.20.2.94.

[38] Wang X., Wang Y., Zhang S., Yao L., Xu S., *Analysis and Prediction of Gestational Diabetes Mellitus by the Ensemble Learning Method*, International Journal of Computatiomal Intelligence Systems, 2022,15(72), DOI: 10.1007/s44196-022-00110-8.

[39] Wei L.L., Pan Y.S., Zhang Y., Chen K., Wang H.Y., Wang J.Y., *Application of machine learning algorithm for predicting gestational diabetes mellitus in early pregnancy*, Frontiers of Nursing, 2021,8(1):209-221, DOI: 10.2478/fon-2021-0022.

[40] Wen L., Li X., Gao L., Zhang Y., *A new convolutional neural network-based data-driven fault diagnosis method,* IEEE Transactions on Industrial Electronics, 2018,65(7):5990-5998, DOI:10.1109/TIE.2017.2774777.

[41] Ye Y., Xiong Y., Zhou Q., Wu Z., Li X., Xiao X., *Comparison of machine learning methods and conventional logistic regressions for predicting gestational diabetes using routine clinical data: a retrospective cohort study,* Journal of Diabetes Research, 2020,4168340:1-10, DOI: 10.1155/2020/4168340.

[42] Zhou F., Ran X., Song F., Wu O., Jia Y., et al., *A stepwise prediction and interpretation of gestational diabetes mellitus: Foster the practical application of machine learning in clinical decision*, Heliyon, 2024,10(1):12(e32709), DOI: 10.1016/j.heliyon.2024.e3270.

[43] Zou Q., Qu K., Luo Y., Yin D., Ju Y., Tang H., *Predicting Diabetes Mellitus With Machine Learning Techniques,* Frontiers in Genetics, 2018,9:515, DOI: 10.3389/fgene.2018.00515.