Researchers Journal of Science and Technology (REJOST)
https://rejost.com.ng

Effective Classification of Diabetes Mellitus Using Support Vector Machine Algorithm

Anietie Ekong*[1], Immaculata Attih[1], Gabriel James[2], Unyime Edet[3]

Received: December 12, 2023 /Accepted: February 16, 2024

Abstract

Diabetes mellitus is one of the metabolic disorders that results in blood glucose levels that are higher than normal. Insulin regulates the flow of sugar from the blood into our cells, where it can be stored or utilized as fuel. Diabetes makes the body unable to create enough insulin or to utilize it effectively. Attempts to classify patients with the disease have been made but these attempts mostly fall short of the necessary diagnostic accuracy and reliability. Early diagnosis and proper management of those with the disease play a crucial role in lowering the burden on patients and can significantly reduce diabetes-related complications and enhance their quality of life. This study aims to develop a simple, non-invasive model to classify diabetes mellitus based on readily recognizable and related risk factors using machine learning. The model was developed using six (6) input variables. Missing value and outlier removal, min-max normalization, feature extraction and feature selection using principal component analysis were methods of data preprocessing applied to the raw dataset. 5-fold cross-validation was used to train and validate the model. Performance evaluation measures included accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AU-ROC). The model development and analysis were carried out using Python 3.9.The classification accuracy, precision, F1-score, and recall for the healthy class were 98%, 97%, 98%, and 98%, respectively while for the diabetic class, 99% precision, 97% recall, 98% F1-score, and 98%accuracy were obtained. This shows that the model can be used to reliably classify diabetes mellitus with high accuracy.

✉ Anietie Ekong: anietieekong@aksu.edu.ng
[1]Department of Computer Science, Akwa Ibom State University, Ikot Akpaden, Nigeria
[2]Department of Computing, Topfaith University, Mkpatak, Nigeria
[3]Department of Computer Science, Akwa Ibom State Polytechnic, Ikot Osurua, Nigeria

Keywords: Diabetes; Machine Learning; Support Vector Machine; Artificial intelligence; Diagnoses; Clinical Decision Support System

1.0 Introduction

Diabetes mellitus is one of the metabolic disorders that results in blood glucose levels that are higher than they should be. Insulin regulates the flow of sugar from the blood into our cells, where it can be stored or utilized as fuel. Because of diabetes, the body is unable to create enough insulin or to utilize it effectively, according to Healthline (2023). Type-2 diabetes is the most common type of diabetes and accounts for around 90% of all diabetes occurrences. Family history, age, obesity, the distribution of body fat, race, gestational diabetes, and lifestyle are some of the prevalent risk factors. According to a study by Feldman *et al*. (2017), increasing dietary fiber consumption lowers the risk of developing diabetes. Additionally, diabetes mellitus can appear as a secondary disorder brought on by a primary illness like pancreatic disease, a hereditary trait like myotonic dystrophy, or medications like glucocorticoids while pregnancy - related gestational diabetes is a transient disorder.

Artificial intelligence (AI) models are valuable tools that can identify diabetes (Orlando *et al.*, 2023). AI techniques can be used to deduce and make clinical decisions by using machine learning (ML) to optimize the knowledge base in order to reveal clinically relevant information (Essien and Umoren, 2021). Diabetes classification can be done using machine learning. In Machine learning, classification uses a specific dataset to model, with the goal of obtaining certain anticipated class labels. The classification requires training a large number of datasets. The seriousness of diabetes has given rise to it becoming a topic for research, with researchers discovering different methods to face the situation. This research aims to develop a machine learning model using Support Vector Machine (SVM) for the efficient classification of diabetes mellitus.

Early detection and diagnosis of those at high risk of developing diabetes will play a crucial role in lowering the burden of this disease. The creation of simple, non-invasive tools to classify diabetes mellitus based on readily recognizable socio-demographic, lifestyle, and physical characteristics can increase awareness of the presence of such a condition on time.

Several studies have been carried out using machine learning for diagnosis. A study by Kumari and Chitra (2013) on the application of SVM in the classification of diabetes Mellitus underlines the importance of ML in clinical decision-support systems. An accuracy of 78% on the dataset used was obtained. The research did not however subject the dataset through the feature subset selection process and this impacted negatively on the accuracy which implies that the system was less reliable as the number of false diagnoses was high.

A precise and quick methodology for diagnosing diabetes mellitus was developed by Pei *et al*. (2019). The method used five standard classifications to categorize patients with diagnoses based on nine non-invasive and easily ascertainable clinical features, including race, age, gender, Body Mass Index (BMI), hypertension, histories of cardiovascular disease, family histories of diabetes, physical activity, job stress, and salt stress. The results showed that the J48 decision tree classification, which is a popular decision tree algorithm that creates decision trees by recursively partitioning data based on attribute values, had the highest value in comparison to other classifiers and may be used to screen patients. In a study by Ekong (2023), an evaluation of the effectiveness of ML algorithms toward the early detection of cardiovascular diseases (CVD) was carried out. SVM, Random Forest (RF), and K-Nearest Neighbour (KNN) were evaluated and KNN emerged as the best model both in training and test performances and was recommended for the early diagnosis of CVD.

This showed how effective ML algorithms could be used in classification.

Ekong and Udo (2023) used an ML-based model for the prediction of fasting blood sugar levels to help in controlling CVD. The General Logistic Model (GLM) was adopted for the prediction of Fasting Blood Sugar levels. Performance analysis results showed effective prediction with an accuracy of 70% accuracy. Maniruzzaman et al. (2020) developed a machine learning-based system for predicting diabetic conditions using Logistic regression (LR) to identify the risk factors. Some classifiers were adopted to predict diabetes in patients using three types of partition protocols. It was concluded that some factors constitute risk factors for diabetes and that with an accuracy of 94.25%, the combined LR-based and RF-based classifier was better than using the algorithms individually. In another study by Ekong *et al.* (2022), a supervised ML model was used for the effective classification of patients with Covid-19 symptoms based Bayesian belief Network. The dataset was gathered from experts' review of symptoms that suspected cases exhibited and it was found to be consistent with reviews of the symptoms that Covid-19 patients exhibit. The model yielded 98% accuracy, showing a significant classification accuracy of patients with Covid-19.

Further study by Miao et al. (2020) was the development of a machine learning-based model to assess the risk of developing CVD due to type-2 diabetes (T2D). The study concluded that ML algorithms were highly effective in the classification of T2D. Abdulkader and Alazzawi (2021), after comparing five Machine Learning Algorithms in the classification of Diabetes concluded that Random forest, with an accuracy of 99%, stood out best among K-nearest neighbor, Support vector machine, linear regression, KNN, SVM, and LR. More so, in a study by Alshari and Odabas (2022), a machine learning model to predict diabetes was built. They used, LightGBM, and CatBoost to develop their model based on health habits and a predictive accuracy of 84.96% was obtained and it was deduced that health habits could be used, to a good extent, to determine the likelihood of someone suffering from type-2 diabetes in the future.

Prasannavenkatesan *et al*. (2022) researched the traditional classification algorithms and neural network-based ML for the diabetes dataset using different performance metrics. They used K-nearest neighbor, multilayer perception, and other ML algorithms to classify and predict patients who suffer or may likely suffer from diabetes later in life.

Nair and Ruqaiya (2021) evaluated the different machine learning algorithms for the classification of diabetes Mellitus to determine which one could best fit that classification need. They concluded that among Logistic Regression, Naive Bayes, SVM, and Decision Trees algorithms, the Naïve Bayes algorithm produced the best accuracy of 79.8%.

All the literature reviewed showed the efficacy of machine learning in disease diagnoses and in the classification of diabetes mellitus, but neither their classification nor predictive accuracy was sufficient to make them sufficiently reliable, a factor that is the most important consideration in clinical decision support systems, or they were not directly addressing the subject of this research. The low classification accuracy in some was partly due to their skipping of important machine learning procedures among other reasons. It is this gap that this research intends to bridge. This study aims at developing a simple, non-invasive model to classify diabetes mellitus accurately, based on readily recognizable and related risk factors using machine learning. The proposed work framework is shown in Figure 1.

2.0 Research Methodology

The main aim of this research is to apply a machine learning model to efficiently classify the type of diabetes mellitus. As shown in Figure 1, it consists of data collection, data preprocessing, feature extraction/selection, and model development. The results of the model classification are evaluated using standard metrics. The feature variables are presented in Table 1.

The Secondary data used comprised the diabetes dataset acquired from the open database of the Hospital Frankfurt, Germany, which contains 2,000 sample points with a total of 9 attributes. The cross-section of the dataset is presented in Table 2

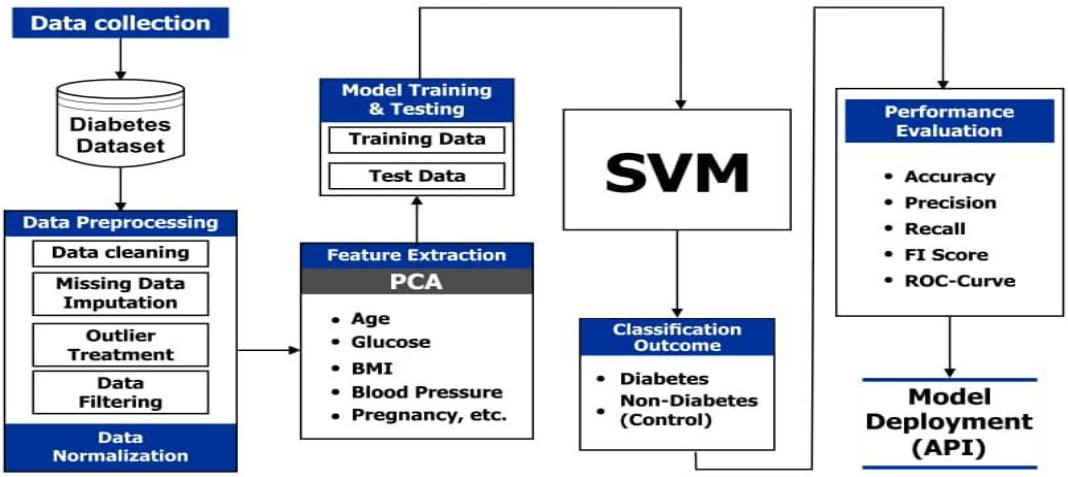

Figure 1: proposed system architecture

Table 1: Features variables

S/N	Attributes	Description	Data Type
1	Age	Age of a Patient	Integer
2	BMI	Body Mass Index	Float
3	Blood Pressure	Diastolic Blood pressure	Integer
4	Skin Thickness	Thickness of Triceps skin fold (mm)	Integer
5	Insulin	2-Hour serum insulin (mu U/ml)	Integer
6	Pregnancies	Number of pregnancies	Integer
7	Diabetes Pedigree Function	Diabetes Pedigree Function	Integer
8	Glucose	An oral glucose tolerance test, plasma glucose concentration was measured after 2 hours.	Integer
9	Outcome	Diabetes =1 Healthy = 0	Categorical

Table 2: Dataset

SN	Pregnancies	Glucose	B.P	Skin Thickness	Insulin	BMI	DPF	Age	Outcome
1	2	138	62	35	0	33.6	0.127	47	1
2	0	84	82	31	125	38.2	0.233	23	0
3	0	145	0	0	0	44.2	0.63	31	1
4	0	135	68	42	250	42.3	0.365	24	1
5	1	139	62	41	480	40.7	0.536	21	0
6	0	173	78	32	265	46.5	1.159	58	0
7	4	99	72	17	0	25.6	0.294	28	0
8	8	194	80	0	0	26.1	0.551	67	0
9	2	83	65	28	66	36.8	0.629	24	0
10	2	89	90	30	0	33.5	0.292	42	0
11	4	99	68	38	0	32.8	0.145	33	0
12	4	125	70	18	122	28.9	1.144	45	1
13	3	80	0	0	0	0	0.174	22	0
14	6	166	74	0	0	26.6	0.304	66	0
15	5	110	68	0	0	26	0.292	30	0
16	2	81	72	15	76	30.1	0.547	25	0
17	7	195	70	33	145	25.1	0.163	55	1
18	6	154	74	32	193	29.3	0.839	39	0
19	2	117	90	19	71	25.2	0.313	21	0
20	3	84	72	32	0	37.2	0.267	28	0
21	6	0	68	41	0	39	0.727	41	1
22	7	94	64	25	79	33.3	0.738	41	0
23	3	96	78	39	0	37.3	0.238	40	0
24	10	75	82	0	0	33.3	0.263	38	0
25	0	180	90	26	90	36.5	0.314	35	1
26	1	130	60	23	170	28.6	0.692	21	0
27	2	84	50	23	76	30.4	0.968	21	0
28	8	120	78	0	0	25	0.409	64	0
29	12	84	72	31	0	29.7	0.297	46	1
30	0	139	62	17	210	22.1	0.207	21	0
31	9	91	68	0	0	24.2	0.2	58	0
32	2	91	62	0	0	27.3	0.525	22	0
33	3	99	54	19	86	25.6	0.154	24	0
34	3	163	70	18	105	31.6	0.268	28	1
35	10	122	78	31	0	27.6	0.512	45	0
36	4	103	60	33	192	24	0.966	33	0
37	11	138	76	0	0	33.2	0.42	35	0

Preprocessing data cleaning, data normalization, correlation analysis, feature analysis, and data filtering are some of the stages employed in this research.

The mathematical formula for standardization of data in dataset can be expressed as:

$$X_{std} = \frac{x - \bar{x}}{standard\ deviation} \quad (1)$$

where x is the data point, $\bar{x}$ is the mean of the feature variable and X_{std} is the standardized value of x. Normalization, which scales the data between a defined range, often between 0 and 1, is another way of standardization. The data was normalized by dividing each data point by the range after removing the minimum value from each data point. Because raw data is frequently complicated and noisy and may not be immediately useful to a machine learning system, feature extraction is crucial and was carried out. The data was converted into a more readable and useful format that can more accurately depict the underlying pattern or structure of the data by extracting features. The precision of the models trained by extracting features from the results is improved by the extraction of characteristics.

The extracted features are represented as:

$$(H_{fset}") = \{H1, H2, H3 \ \ldots\ldots\ldots\ldots HS\} \quad (2)$$

where H_{fset}" is the feature set that has been extracted and HS is the number of features that have been extracted from the dataset. The features extracted are as contained in Table 2.

One of the steps taken was to plot the glucose and age feature variables using a scatter plot as shown in Figures 2 and 3. The aim of this was to segment the range of healthy people. From the plot, it was determined that healthy people concentrated where their ages were less than 30 and their glucose levels less than 120. The red box in the diagram indicates the range of concentration of the data point for the newly extracted features, which is defined as H1.

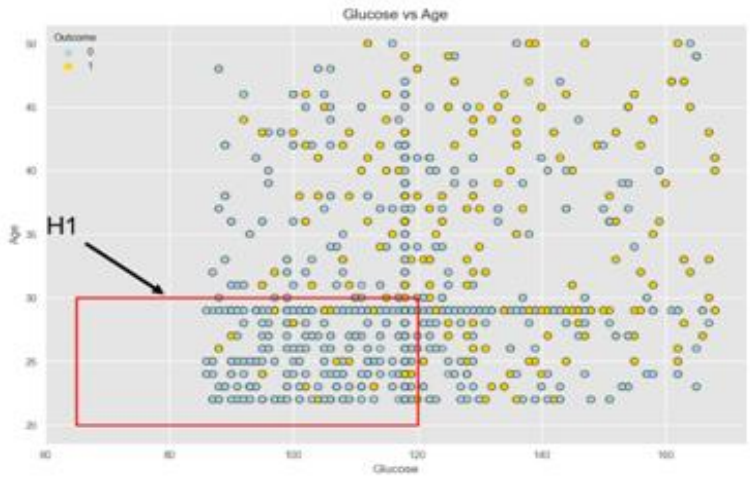

Figure 2: H1 extracted features

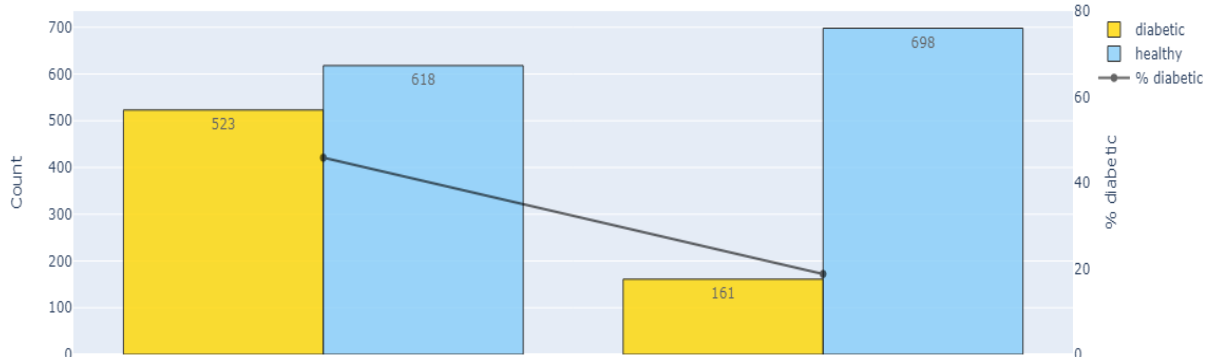

Figure 3: H1: Glucose <=120 and Age <= 30

The line in Figure 3 indicates the diabetic percentage and shows that 18.7% of the H1 newly extracted feature is diabetic, while 81.3% of the H1 feature (that is, data points

having glucose <=120 and age <=30) are healthy. The 18.7% (161 cases of diabetes) translates to 23.5% of the total diabetic cases, while the 45.8% (698 cases of healthy) translates to 47% of the total number of healthy cases. The diagram illustrating the distribution of the newly extracted feature H1 in terms of the total number of diabetic and healthy classes is shown in Figure 4.

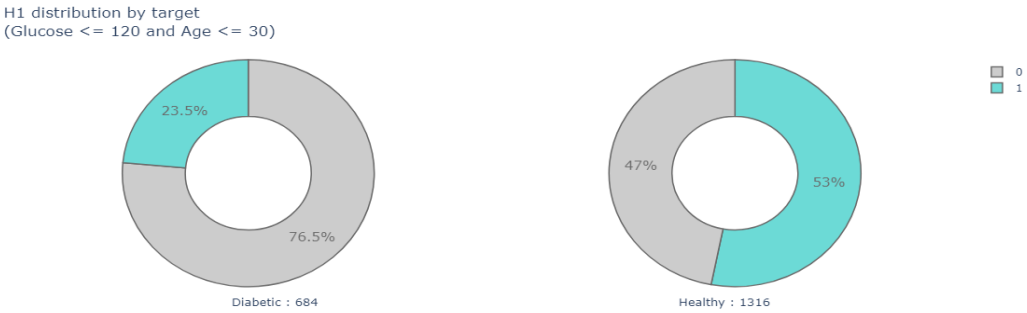

Figure 4: H1 distribution by target variable

Also, the BMI is another criterion used for early diagnosis of diabetes expressed mathematically as:

$$BMI = \frac{W}{H^2} \qquad (3)$$

where H is the height in meters and w is weight in kilograms. According to World Health Origination (2021), BMI is defined as the ratio between the weight and the square of the height of a person and the unit is expressed in kg/m^2. When the BMI of a person is above 30, the individual is considered obese. Therefore, in our data extraction, a new feature H2 was derived from the dataset. H2 features represent the class of diabetic and non-diabetic whose BMI is below, 30 as shown in Figure 5. Similar method of data extraction was used for extracting H0, H3, H4, H6, H7, H8, H9, H10, H11 and H14 new features from our dataset. Furthermore, after the data extraction, a correlation matrix was plotted to indicate how each of the feature sets `correlates to each other as shown in Figure 6

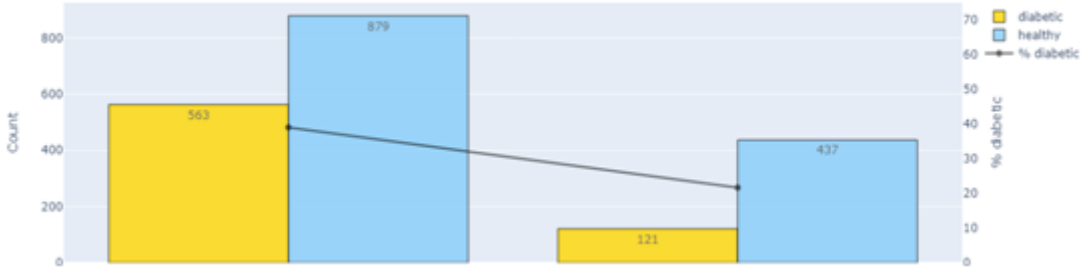

Figure 5: H2 distribution by target variable

3.0 Model Formulation

Machine learning classification involves developing a prediction model that can accurately identify whether a patient has diabetes or not with high accuracy. The model uses a set of input features to make a classification or prediction about whether the patient has or will have diabetes. The model is trained on a dataset of patients with and without diabetes, with their corresponding clinical data, to learn patterns and relationships between the features and the disease.SVM is one of the supervised learning methods used for classification in medical diagnoses. SVM minimizes the empirical classification error while maximizing the geometric margin. By

implicitly translating their inputs into high-dimensional feature spaces, the kernel technique enables SVMs to do non-linear classification successfully. When given a set of points from one of the two classes (diabetic or healthy), an SVM, for instance, examines a hyperplane with the highest proportion of points from the same class on the same plane. The best-separating hyper plane minimizes the likelihood of misclassifying test dataset instances while increasing the distance between the two parallel hyper planes. The optimization objective of our SVM model is to maximize the distance separating the decision boundaries (hyper plane) between diabetic and healthy classes (margin). The data samples that are close to the hyper plane are called support vectors as shown in Figure 7.

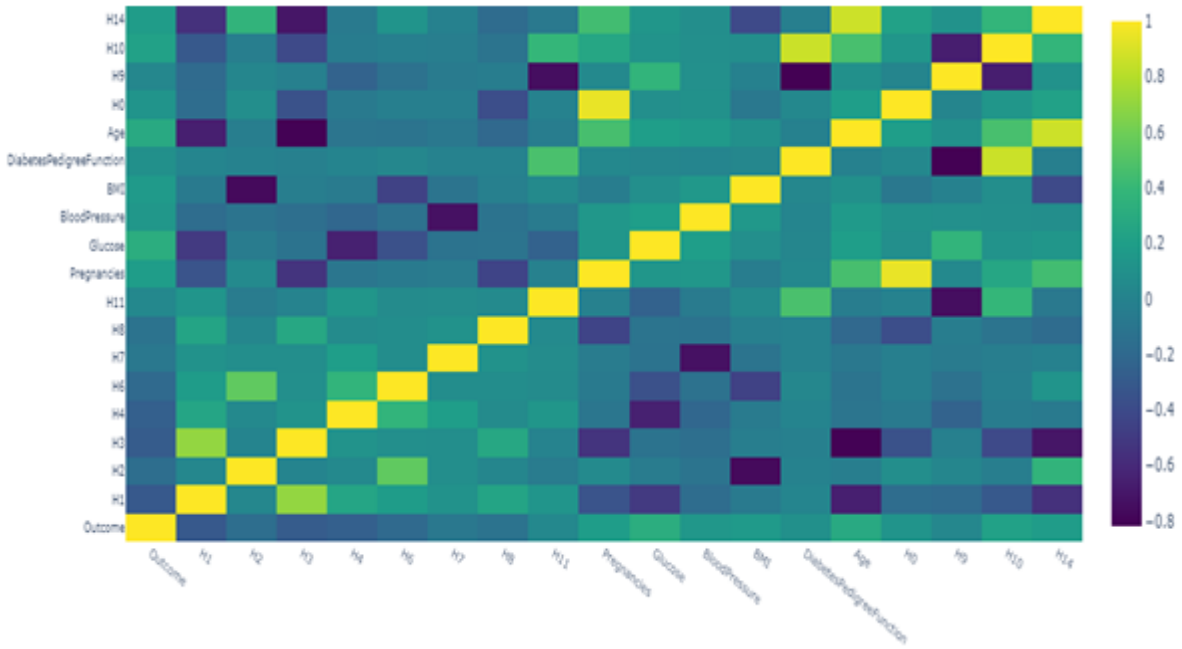

Figure 6: Correlation Matrix after Feature Extraction

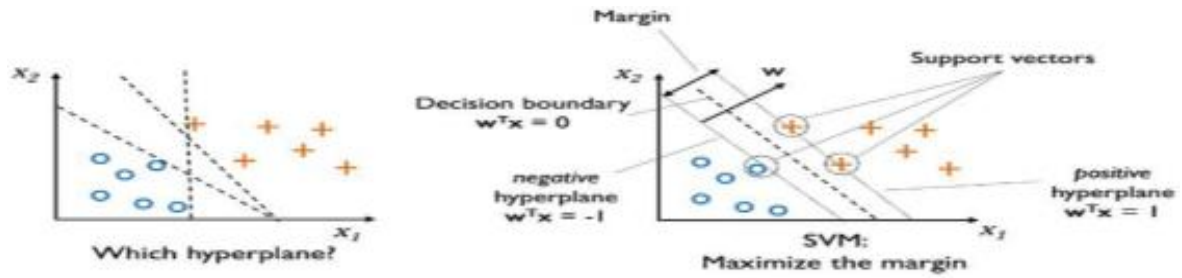

Figure 7: General classification representation of SVM hyper plane

In Figure 7, diabetic is represented as positive support vectors while the negative stands for healthy class, and both classes are parallel to the hyper plane and represented as:

$$w_o + w^T x_{pos} = 1 \qquad (4)$$

$$w_0 + w^T x_{neg} = -1 \qquad (5)$$

where w is a scalar, w_0 is a g-dimensional weight vector and x_{pos}&x_{neg} are the diabetic and healthy feature data sample vectors, respectively.

Subtracting equation 5 from equation 4, we have:

$$\rightarrow w^T(x_{pos} - x_{neg}) = 2 \qquad (6)$$

Equation 6 can be normalized using the length of the vector w as follows:

$$||w|| = \sqrt{\sum_{j=1}^{m} w_j^2} \qquad (7)$$

The following equation is obtained after normalization:

$$\frac{w^T(x_{pos} - x_{neg})}{||w||} = \frac{2}{||w||} \qquad (8)$$

The left side of equation 9 can be interpreted as the distance between the diabetic and healthy hyper plane which is the margin we intend to maximize. Therefore, the objective of the SVM is to maximize the margin by maximizing $\frac{2}{||w||}$ under the constraints that each of our data samples is classified correctly and can be represented in the following formulas:

$$w_0 + w^T x^i \geq 1 \; if y^i = 1 \qquad (9)$$

$$w_0 + w^T x^i \leq -1 \; if y^i = -1 \qquad (10)$$

For $i = 1 \ldots where$ N is the total number of datasets sampled.

These two equations imply that all the healthy classes should fall on only one side of the negative hyperplane while all the diabetic cases will fall behind the positive side of the hyperplane, which can be expressed as:

$$y^i(w_0 + w^T x^i) \geq 1 \qquad \forall_i \qquad (11)$$

More so**,** principal component analysis being an unsupervised learning technique for reducing the dimensionality of data, increasing interpretability, and at the same time minimizing information loss based on the correlation between feature sets, aids in the discovery of patterns in the data. The extraction of new features from the dataset yielded 18 feature sets. To use PCA to reduce the dimensionality of the dataset, we construct a d *X* h-dimensional transformation matrix $\boldsymbol{T}$**,** that will enable the mapping of a vector x of the features of the training sample unto a new h - dimensional feature subspace with much lower dimensions when compared with the original d-dimensional feature space. This can be shown in the following equation:

$$x = [x_1, x_2, \ldots, x_d], \quad x \in \mathcal{R}^d \qquad (12)$$

Transformation matrix $\boldsymbol{T}$ is used to transform equation 12 as $\boldsymbol{T} \in \boldsymbol{\mathcal{R}}^{dxh}$ as follows:

$$\boldsymbol{xT} = \boldsymbol{z} \qquad (13)$$

Therefore, this results in a vector$\boldsymbol{z} = [z_1, z_2, \ldots z_h], \quad \boldsymbol{z} \in \boldsymbol{\mathcal{R}}^h$. Since we transformed our original d-dimensional data onto the new $h - dimensional$ subspace ($h \ll$ d), therefore the first orthogonal axes or the principal components will have the highest variance. Given the restriction that these components are uncorrelated (orthogonal) with the other principal components, all succeeding principal components will have the maximum variance. Even if the input characteristics are correlated, the output principal components will be mutually orthogonal (uncorrelated). The approach that was used for the implementation of the PCA algorithm can be summarized as follows:

i. standardized the original d-dimensional dataset
ii. construct the covariance matrix and decompose it into its eigenvectors and eigenvalues
iii. sort the eigenvalues in decreasing order corresponding to the rank of the eigenvectors
iv. select the h eigenvectors, which corresponds to the h largest eigenvalues where h is the new dimension of the new feature space.
v. construct a projection matrix T, from the top h eigenvectors and

vi. transform the d-dimensional original input dataset, X, using T to get the new h-dimensional feature subspace.

After data extraction, the dataset was standardized using a simple Python standard scaler function. The second step involved the construction of a covariance matrix. The pairwise covariances between the various features are stored in the symmetric d×d-dimensional covariance matrix, where d is the number of dimensions in the dataset. The covariance between two feature sets x,_a and x_b in a population can be calculated using the following formula:

$$\sigma_{ab} = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_a^i - \mu_a\right)\left(x_b^i - \mu_b\right) \quad (14)$$

where μ_a and μ_b are the sample means of the features a and b, respectively. Additionally, a positive covariance between the two features simply indicates that the two features decrease or increase together while a negative covariance indicates that the two features change in opposite directions to each other. A covariance matrix of say three features of a dataset can be expressed as follows:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{bmatrix} \quad (15)$$

The eigenvectors of the matrix in equation 15 represent the principal components, that is, the direction of the maximum variance, while the corresponding eigenvalues define the magnitude of the vectors. Since our datasets after data extraction resulted in 18 feature variables, it therefore means that we will obtain 18 eigenvectors and eigenvalues from the 18 ×18-dimensional covariance matrix. The third step is to solve for the eigenvalues and eigenvectors for the covariance matrix using the following expression:

$$\Sigma \boldsymbol{v} = \boldsymbol{\lambda v} \quad (16)$$

where λ is a scalar: the eigenvalue. Manual computation of eigenvalues for 18 x18 matrix is a very tedious and daunting task. Therefore, we opted to use a simple Python library known as $linalg.eig$ function to compute the eigenvalues and eigenvectors. The covariance matrix decomposition yielded a vector consisting of 18 eigenvalues and corresponding eigenvectors stored in the column as 18 × 18 matrix.

The portion of the eigenvectors (principal components) that includes the majority of the information (variance) needed to minimize the dimensionality of our dataset by compressing it onto a new feature subspace. We ordered the eigenvalues by decreasing magnitude since the eigenvalues determine the magnitude of the eigenvectors. Based on the magnitudes of their corresponding eigenvalues, we are interested in the top h eigenvectors. To determine the majority of the information, that is, the variance, we had to first analyse the explained variance ratio. The ratio is calculated simply by dividing each eigenvalue with the total sum of the eigenvalues as expressed as follows:

$$Explained\ variance\ ratio = \frac{\lambda_i}{\sum_{j=1}^{d}\lambda_i} \quad (17)$$

The result of the explained variance ratio presented here indicates that the first eight principal components account for all the variance in the dataset, while the first four principal components account for more than 75% of the variance as shown in figure 8.

Now the 2000 samples of our dataset were then transformed from a 2000 x18-dimensional dataset onto eight principal components using$\boldsymbol{X' = XT}$. However, for data visualization and clustering, the first three principal components of the transformation were considered, which account for more than 50% of the variance ratio.

```
pca.explained_variance_ratio_
✓ 0.0s
array([0.28814874, 0.22757666, 0.12651918, 0.12486507, 0.08134822,
       0.0759775 , 0.016887  , 0.01467971, 0.00989071, 0.00778553,
       0.00684235, 0.00544691, 0.00473883, 0.00316785, 0.00240787,
       0.00181761, 0.00139555, 0.00050469])
```

Figure 8: Explained variance ratio

4.0 Results and Discussion

To avoid overfitting the model, cross-validation was used to validate portions of the training data directly. The primary objective of building a model is to create a generalized model based on the present data that will excel when applied to future data. However, we need to simulate that unseen data using the data we already know to evaluate a model's performance on that data. To do this, we divide the data we have into training and testing sets.

The same approach to fine-tuning our algorithm can be applied. We can assess how well certain model hyper parameter settings perform on the test set. Retrain the model using a new training and test set partition and new hyper parameter values. If the new parameters outperform the old ones, we take them and keep going through the same procedure until we have the hyper parameters set to their ideal values. The cross-validation process is shown in Figure 9.

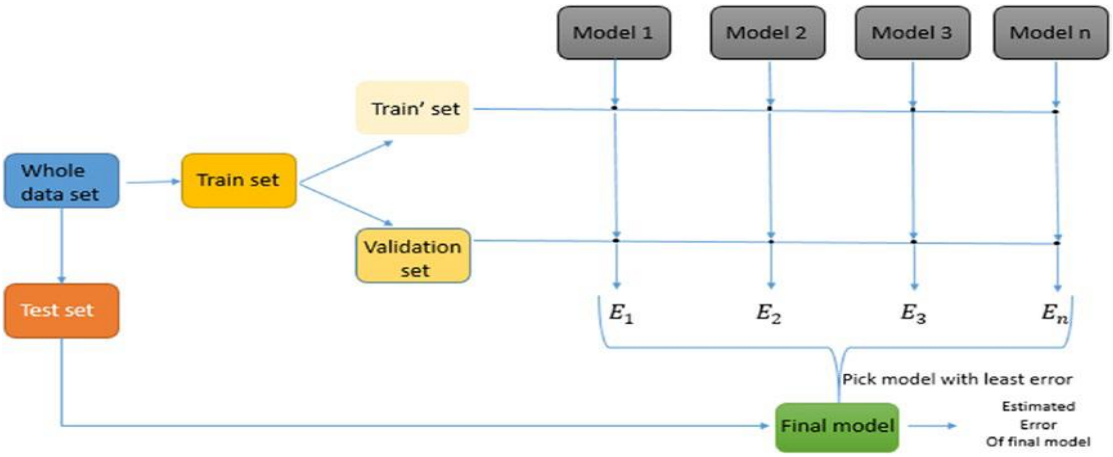

Figure 9: Cross-validation process for model building and tuning

The dataset is divided into training and a test set. The training dataset is further split into actual training data and a validation dataset, while the test dataset is kept completely separate from the entire learning process. In the diagram, the models listed as models 1 to n are single-type support vector models with different hyper parameters. After the models were created, how well they performed on the validation set was evaluated and the model with the best results was chosen model assessment metrics of accuracy, f1 Score, precision and recall were used.

The need for cross-validation stemmed from the fact that by dividing the dataset into test and validation sets, we lost a large amount of data, which could have been used in the training and modeling process of our model. Again, if the model's error is evaluated based on a single iteration, this will result in a serious mistake;

instead, we intended to find the average error of the model by training multiple iterations of the same model. On the other hand, if this is done on the same dataset, it will drastically reduce the training error. However, the testing error will increase thereby resulting in little or no improvement in the model's performance. To address these two issues, we employed the technique of Cross-validation. The stratified k-fold cross-validation strategy is an improved k-fold strategy whereby the training dataset is split into k-subsets, such as k=5, as used in this study. Out of this k subset, the model k-1 subsets were trained while the remaining one subset is used for validation. The error was averaged over the k models that were produced by creating various iterations of the model after the training was done in k times. In each of these cycles, we keep switching the validation set, ensuring that the model gets trained on a new subset of data each time.

4.1 Model Performance Evaluation Metrics

Typical metrics used for evaluation of classification type of problems were applicable and used in this work and are explained as follows:

1. **Accuracy:** This represents the number of correct predictions made by the model divided by the total number of predictions. It is expressed mathematically as:

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \quad (18)$$

2. **Precision and recall:** These metrics are used when the goal is to minimize false positives or false negatives. Precision is the ratio of true positives to the total number of predicted positives, while recall is the ratio of true positives to the total number of actual positives. The mathematical representation of the precision and recall are expressed in the following equations:

$$Precision = \frac{T_P}{T_P + F_p} \quad (19)$$

$$Recall = \frac{T_p}{T_n + F_p} \quad (20)$$

3. **F1 score:** This is a weighted average of precision and recall, and is commonly used when the goal is to balance the trade-off between precision and recall. It is expressed as:

$$F1_{score} = 2 \times \frac{precision \times ecall}{precision + recall} \quad (21)$$

4. **AUC-ROC:** This metric is used to evaluate binary classification models and represents the area under the receiver operating characteristic curve. It measures the model's ability to distinguish between positive and negative classes.

5. **Confusion Matrix:** The number of precise and unreliable predictions is categorized in a table called the confusion matrix. The effectiveness of a classification model when applied to a set of test data for which the true values are known is usually summarized in a confusion matrix. True positives and negatives are included in the matrix. The effectiveness of a classification model when applied to a set of test data for which the true values are known is usually summarized in a confusion matrix. The confusion matrix table reveals how many relevant samples the classifier has identified and how many of them have been accurately identified.

4.2 Model Evaluation Result of Cross-Validation

The following results were obtained during the building process of the model using the cross-validation technique. The result obtained is presented in Table 3below. The results cover all the metrics used in the evaluation of classification problems, ranging from the classification report, confusion matrix, and ROC curve. It is important to note that this result was evaluated using the validation data from the cross-validation process. The

classification report of the cross-validation for five folds during the model training is presented in Table 3.

The Confusion matrix evaluated using the cross-validation data is illustrated in Figure10 and the Receiver Operating Characteristic (ROC) evaluated on the cross-validation data for five folds is shown in figure 11. The brown colour in Figure 11 represents the healthy class and the pink colour represents the diabetic class during validation.

Table 3: Fivefold cross validation

Cross Validation - 5 folds
Support Vector

Fold	Accuracy	Precision	Recall	F1 score	Roc auc
1	0.955	0.944	0.966	0.955	0.965
2	0.962	0.945	0.981	0.962	0.975
3	0.936	0.925	0.947	0.936	0.943
4	0.948	0.916	0.986	0.949	0.967
5	0.969	0.954	0.986	0.969	0.984
mean	0.954	0.937	0.973	0.954	0.967
std	0.011	0.014	0.015	0.011	0.014

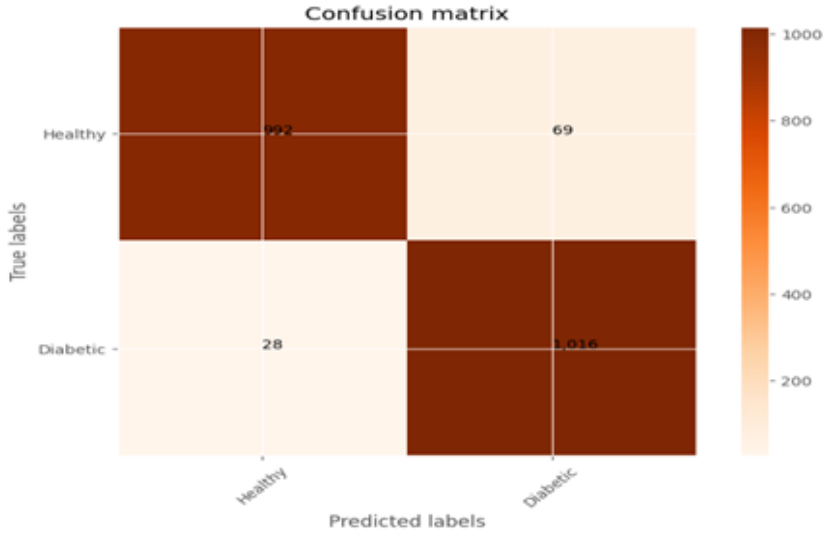

Figure 10: Confusion matrix evaluated using the cross-validation data

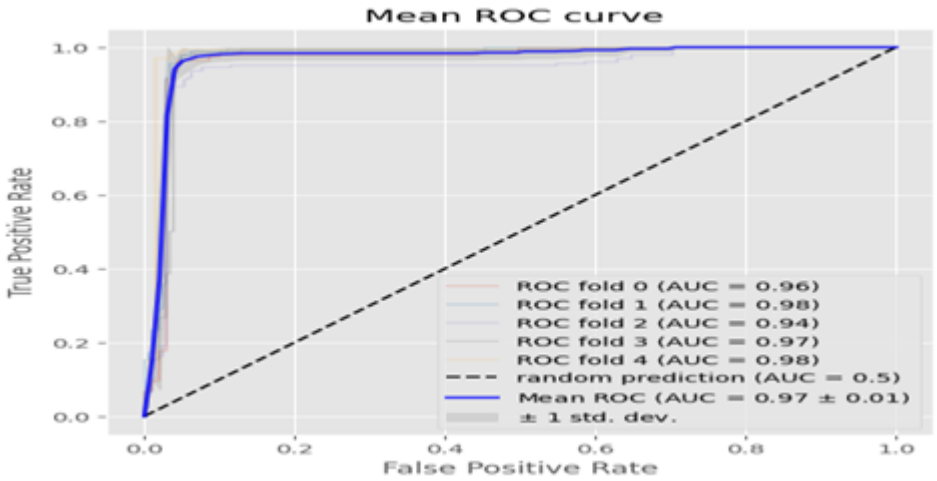

Figure 11: Receiver Operating Characteristic (ROC) evaluated on the cross-validation data for five folds

The following results were obtained when the hybrid model was evaluated on the test data. The results were evaluated using the metrics discussed in the previous section. The summary of the classification report which contains the accuracy, precision, f1-score, and recall values are presented in Table 4. The confusion matrix obtained during the evaluation of the project model is presented in Figure 12. The deep blue colour in Figure 12 represents the healthy class and the light blue colour represents the diabetic class during testing. It is based on the test dataset consisting of 527 sample size, 253 out of 255 were correctly predicted as healthy while only 2 samples were misclassified as diabetic. In the same vein, 265 cases of diabetes were correctly classified while only 7 cases were misclassified as healthy. This result further proves that our model performed extremely well on the test data. Comparing the result obtained using the test data and validation data during training; it is obvious that there was no issue of over fitting or under fitting. Also, it is important to note that due to the extensive approach of data preprocessing used; we can observe the similarity of the performance of our model both during training and testing, indicating that the model is reliable, valid, and can be generalized for prediction of unseen data.

Table 4: Classification Report for Test Dataset

Class	Precision	Recall	F1-Score	Accuracy
Healthy	0.97	0.99	0.98	0.98
Diabetic	0.99	0.97	0.98	

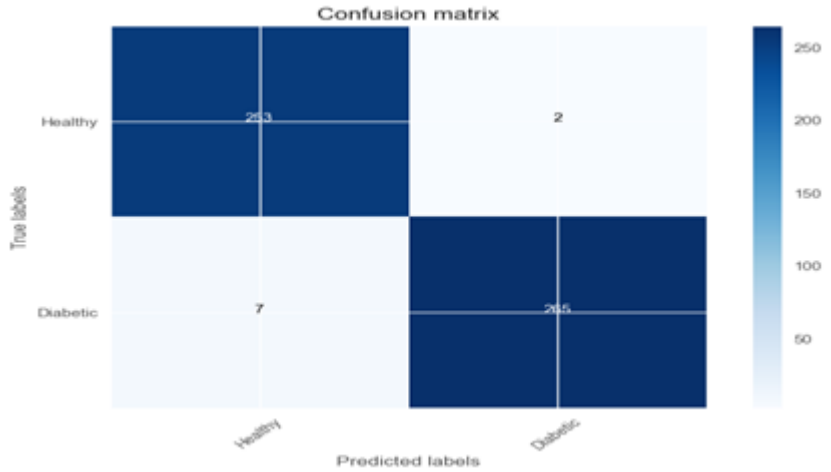

Figure 12: Confusion matrix for Test data

The ROC curve evaluated using the test dataset is shown in Figure 6. The y-axis indicates the true positive rate while the false positive rate is indicated on the x-axis. On the bottom right corner of the ROC curve is the AUC score of 0.98. The result of the classification report presented in Table 1 indicates the performance of the model evaluated using the traditional and acceptable metrics of assessing machine learning classification problems. The result was evaluated using the test (unseen) dataset, which was split and reserved from the original data. The test data was not involved in the training process of the model; instead, a cross-validation technique was used in building the model. After the test data was used to evaluate the performance of the model, 0.97 and 0.99 precision scores were obtained for healthy and

diabetic classes, respectively, while 0.99 and 0.97 scores were obtained for recall for healthy and diabetic classes, 0.98 scores was recorded as the F1 score for both healthy and diabetic classes and the overall accuracy of model was recorded as 0.98. This implies that 97% and 99% of the time our model is able to precisely classify healthy and diabetic classes, respectively, while the model was able to predict accurately, 99% and 97% of the healthy and diabetic classes. The 98% F1 score of our model indicates the average score of how precise our model is and its recall. Finally, a 98% accuracy score obtained from the test dataset indicates that the model was able to make 98% accurate predictions. The ROC curve affirmed the earlier conclusion observed in the results obtained from the classification report and confusion matrix that the model performed extremely well on the test dataset, as shown in Figure 13. The area of the curve is close to 1 (0.98), indicating that the model will predict accurately 98% of the time. This further implies that our model performance is better, than similar studies that have been conducted.

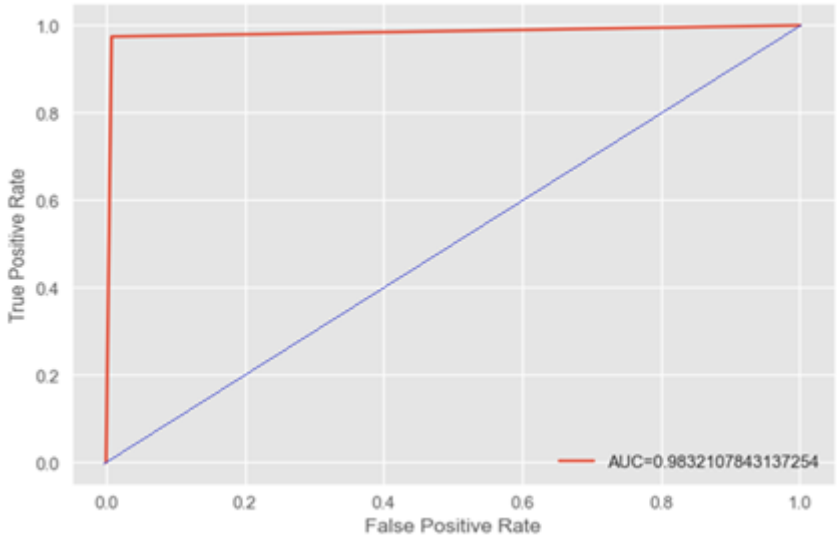

Figure 13: ROC curve based on the test dataset

5.0 Conclusion

Despite efforts by researchers to classify diabetes mellitus manually and then using Machine Learning algorithms, they still fell short of expectations as their accuracies were mostly low and inefficient making such models to be unreliable classifiers. This study aimed at developing a simple, non-invasive, and highly accurate model to classify diabetes mellitus based on readily recognizable and related risk factors using a machine learning algorithm. Support vector machine which is one of the ML algorithms was developed for this task. The steps in the classification of diabetes include data collection, preprocessing, feature extraction, and classification. The performance of the proposed work was analyzed by first using the cross-validation technique for the training model in combination with random search. Standard classification evaluation methods were applied to the training phase of the model while analysis was performed using the validation data. 95% accuracy, 93% precision, 97% recall, 95% F1 score, and 95% AUC score were obtained from the validation data during the training phase. After the training phase of the model was completed, test data was used for the performance evaluation of the model. Classification accuracy, precision, F1-score, and recall for the healthy class were 98%, 97%, 98%, and 98 %, respectively while

for the diabetic class, 99% precision, 97% recall, 98% F1-score, and 98 % accuracy were obtained showing greater efficiency over to existing work.

References

Abdulkader, R. M. & Alazzawi, A. K. (2021) A Comparison of Five Machine Learning Algorithms in the Classification of Diabetes Dataset, *Turkish Journal of Computer and Mathematics Education,* 12(14), 1244 – 1259

Alshari, H., & Odabas, A. (2022). Machine Learning Model to Diagnose Diabetes Type 2 Based on Health Behavior**,**Gazi University Journal of Science35 (3),834-852.

Ekong, A. (2023). Evaluation of Machine Learning Techniques towards Early Detection of Cardiovascular Diseases. *American Journal of Artificial Intelligence*. Vol.7, No. 1, 2023, pp. 6-16.

Ekong, A., Ekong, B. & Edet, A. (2022). Supervised Machine Learning Model For Effective Classification of Patients With Covid-19 Symptoms Based On Bayesian Belief Network, *Researchers Journal of Science and Technology,* 2: 27 – 33.

Ekong, A., Udo, E. (2023). Machine Learning based Model for the Prediction of Fasting Blood Sugar Level towards Cardiovascular Disease Control for the Enhancement of Public Health *International Journal of Computer Applications (0975 – 8887),* Volume 184 – No. 52.

Essien V., Umoren I., Umoh I. (2021). A Bio-Informatics System for Intelligent Classification of Severity Index of Hypertension. International Journal of Innovative Research in Sciences and Engineering Studies. 1(2) :1-8

Feldman, E. (2017). New Horizons in Diabetic Neurpathy: Mechanisms, Bioenergetics, and Pain *Science Direct Journal*, Volume 93, Issue 6, pages 1296-1313.

Healthline (2023). *Diabetes*, Retrieved January from https://www.healthline.com-/health/diabetes. Access Date: 1.01.2023

Kumari V. & Chitra R. (2013). Classification of Diabetes Disease Using Support Vector Machine. International Journal of Engineering Research and Applications. 3(2):1797-1801

Maniruzzaman, M., Rahman, M. J., Ahammed, B., &Abedin, M. M. (2020). Classification and prediction of diabetes disease using machine learning paradigm. *Health information science and systems*, 8(1), 1-14.

Miao, L., Guo, X., Abbas, H. T., Qaraqe, K. A., &Abbasi, Q. H. (2020). Using Machine Learning to Predict the Future Development of Disease. In 2020 *International Conferenceon UK-China Emerging Technologies (UCET) (pp. 1-4). IEEE.*

Nair U. and Ruqaiya K. (2021). Classification of Diabetes using Machine Learning. International Conference on Computational Performance Evaluation (ComPE) North-Eastern Hill University, Shillong, Meghalaya, India.

Orlando I., Karina E., Rosalynn O. and Michael C. (2023). Application of Machine

Learning Models for Early Detection and Accurate Classification of Type 2 Diabetes. *Diagnostics* 13 (14).

Pei D., Gong, Y., Kang, H., Zhang, C., & Guo, Q. (2019). Accurate and rapid screening model for potential diabetes mellitus, *BMC Medical Informatics and Decision Making*, 19(1),41.

Prasannavenkatesan T., Usha R., Vidya J. (2022). Diagnosis and Classification of the Diabetes Using Machine Learning Algorithms. Computer Science, 4 (1).

World Health Organization. (2021). Obesity and overweight. https://www.-who.int/news-room/fact-sheets-/detail/obesity-and-overweight