IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 29, 2021, accepted September 9, 2021, date of publication September 29, 2021, date of current version October 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3116383

Memory-Efficient, Accurate and Early Diagnosis of Diabetes Through a Machine Learning Pipeline Employing Crow Search-Based Feature Engineering and a Stacking Ensemble

SHIRINA SAMREEN, (Member, IEEE)
Department of Computer Science, College of Computer and Information Sciences, Majmaah University, Al Majmaah 11952, Saudi Arabia
e-mail: s.samreen@mu.edu.sa

Shirina Samreen would like to thank Deanship of Scientific Research at Majmaah University for supporting this work under Project Number R-2021-224.

ABSTRACT The early diagnosis of diabetes helps in avoiding the major risks associated with the disorder. The proposed research involves the design of a machine learning pipeline which generates the most representative feature subset of minimal size that predicts the onset of Diabetes with highest accuracy. It employs a novel diabetes dataset which is gender-neutral and representative enough unlike the well-known PID dataset. The machine learning pipelines involve multiple feature engineering pipelines to generate a reduced feature subset which is fed into multiple heterogeneous classifiers. The feature engineering involves feature selection as well as feature extraction. The former uses the ANOVA filter and Crow Search Optimization algorithm. The latter employs the Singular Value Decomposition. The classification is performed on the preprocessed dataset using a wide range of heterogeneous classifiers like Naive Bayes', Logistic Regression, K-Nearest Neighbor, Decision Trees, Support Vector Machine, Random Forest, AdaBoost, and GradientBoost as base learners followed by their stacking ensemble. The performance evaluation of each machine learning pipeline is done through Repeated Stratified K-fold Cross Validation using the metrics of accuracy, precision, recall, F1 Score and area under Receiver Operating Characteristic curve. For each pipeline, the number of features in the preprocessed dataset varies and the highest accuracy of 98.4% is achieved with Crow Search algorithm through a stacking ensemble of multiple heterogeneous classifiers. A comparative analysis with a recent related work on the same dataset shows that the proposed feature engineering pipelines with the same set of classifiers outperform with improved accuracy using a feature set of reduced size.

INDEX TERMS Early diabetes diagnosis, machine learning, stacking ensemble, feature engineering, dimensionality reduction, crow search algorithm, ANOVA filter, singular value decomposition.

I. INTRODUCTION

A very common chronic disorder prevalent in the modern world is Diabetes Mellitus. It has become a serious health issue throughout the world irrespective of geographic boundaries. The disorder is associated with the insulin hormone produced by the pancreas. It occurs in one of the following forms: Type 1, Type 2 and Gestational diabetes [1]. Type 1 diabetes is caused when the body's immune system causes the destruction of beta cells of the pancreas. Consequently, the body has deficient insulin which makes the glucose absorption in the bloodstream difficult. Type 2 diabetes happens when the body fails in producing insulin or becomes insulin resistant. This is attributed to various factors like obesity, unhealthy lifestyle and hereditary transmission. Gestational diabetes happens due to hormonal changes in the body during pregnancy. It was found that Diabetes can have its existence up to 7 years prior to clinical diagnosis [2]. The consequences of undetected and untreated diabetes can be quite serious which include damage of vital organs like eyes, kidneys and heart, foot ulcers, pancreatic beta cells destruction, weight loss and sexual dysfunction. These complications tend to be less severe when the period between the onset of the disease and the initiation of the treatment is as small as possible [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

A global survey in 2017 [4] has revealed that 451 million people around the world are already diagnosed with diabetes and it is predicted to rise to 693 million by the year 2045. Yet another study [5] shows that diabetes among people over 18 years has become more prevalent in second and third world countries with a rise in the growth rate of 4.7% to 8.5% from 1980 to 2014. Another research [2] has computed the total annual expenses of a whopping amount of $700 million incurred by Australian healthcare system without the early detection of type 2 diabetes. For United States of America, the year 2017 has incurred an expense of $327 billion [6] for the diagnosed diabetic cases. A global statistical study [7] has found that the year 2011 had 9% of China's population, 8% of India's population and 10% of Bangladesh's population diagnosed with diabetic related complications. Looking at the statistics, it is very clear that bearing the annual expenses for low and middle income countries would be more challenging. Hence it is very imperative that the mechanisms for the early detection of diabetes are in place which could improve the lifespan of the people as well as help in reducing the nation's expenditure towards healthcare.

The most popular diagnosis methods for diabetes are Oral Glucose Tolerance Test and Hemoglobin A1c (HbA1c) test recommended by most physicians upon the observation of symptoms of the disorder. A number of practical difficulties are associated with performing these tests as they are not easily affordable by all income groups as well as time consuming and prone to human errors by the lab technicians. A more cost effective as well as accurate approach towards the diagnosis of diabetes would be through machine learning(ML). In this approach, the historical clinical data related to the patients who were diagnosed positively/negatively with diabetes is needed. Using this data, a ML model is created to predict whether a new patient has the risk of diabetes by looking into the data about the symptoms. Keeping this in mind, we employ a diabetes dataset [8] of 520 instances with 16 attributes which include the major symptoms of the disorder.

The main contribution of this research is to design an optimal machine learning pipeline on a novel diabetes dataset which maximizes the classification accuracy with minimal features. The pipeline design includes data preparation along with various feature engineering techniques to generate a reduced feature subset which is processed through multiple heterogeneous classifiers including the Naive Bayes' (NB) Webb *et al.* [9], K-Nearest Neighbor (KNN) by Cunningham *et al.* [10], Logistic Regression(LR) Tabaei *et al.* [11], Decision Tree (DT) Jenhani *et al.* [12], Support Vector Machine(SVM) Cortes *et al.* [13], Gradient Boost, Adaboost(AB) by Kégl [14], and Random Forest (RF) by Breiman [15]. Apart from this, a stacking ensemble comprising the said classifiers (except RF) as base learners and the RF as a meta learner is also employed to assess the accuracy. The data preparation involves one-hot encoding to transform the categorical features into numeric ones. This is followed by data standardization using standard scaler. The transformed dataset is subjected to the feature engineering step which involves the usage of feature selection and feature extraction techniques to reduce the size of feature subset.

Extensive experiments involving different combinations of feature engineering pipelines with each of the classifiers along with their stacking ensemble are performed so as to minimize the size of feature subset and maximize the classification accuracy.

The paper has been organized in the following manner: Section II covers the literature survey. Section III covers the dataset description, proposed predictive model and metrics used for performance evaluation. Section IV includes detailed analysis of the experimental results along with the ROC curves obtained with different predictive models employing different classifiers with optimal feature engineering pipelines. The conclusion and future work have been presented in section V.

II. LITERATURE SURVEY

A great deal of research has been carried out in recent times towards accurate diagnosis of diabetes using the different ML models upon different datasets. We categorize the existing approaches based upon the criteria of the usage of feature selection methods. It is employed during data preprocessing so as to perform dimensionality reduction to improve the accuracy and reduce the feature space thereby decreasing the storage space requirements.

A. PREDICTIVE MODELING OF DIABETES WITH NO FEATURE SELECTION

An ensemble based approach along with wrapper based feature selection method was proposed by Li *et al.* [16]. wherein a combination of SVM, Artificial Neural Network(ANN) Reinhardt *et al.* [17], and NB classifiers were used for diabetes diagnoses. The method was based upon weighted voting by assigning the weights to each classifier according to their history of making accurate predictions. The dataset used was Pima Indian Diabetes (PID) dataset and the method achieved an accuracy of 80%.

A classifier based upon Genetic Programming was proposed by Pradhan *et al.* [18] using a simplified function pool along with different selection methods for diabetes prediction. An optimal result with a maximum accuracy of 89% was achieved with tournament selection method.

A predictive model for diabetes prediction was developed by Naiarun *et al.* [19] using various classification models including DT, ANN, LR, and NB along with combination of Bagging and Boosting techniques. Apart from these, RF algorithm was also employed and the best classification accuracy of 85.5% was achieved with RF. The dataset was collected from 26 Primary Care Units in Sawanpracharak Regional Hospital during 2012 – 2013.

Diabetes prediction using the data from Canadian Primary Care Sentinel Surveillance network was done by Perveen *et al.* [20]. The method used the AB and Bagging ensemble techniques with J48 (C4.5) Decision Tree as a base

learner along with standalone J48 algorithm. The method involved no data preprocessing steps and the experimental results prove that AB ensemble method has highest performance in comparison with bagging and J48 decision tree.

A classification method employing the Gaussian Process Classification (GPC) by Belhouari *et al.* [21] using the linear, polynomial and radial basis kernel was proposed by Maniruzzaman *et al.* [22] for the prediction of diabetes using the PID dataset. The usage of Gaussian Process (GP) method was attributed to non-linearity and inherent correlation within the medical datasets. The performance was compared with other methods including the Linear Discriminant Analysis (LDA) by McLachlan *et al.* [23], Quadratic Discriminant Analysis (QDA) by Cover [24], and NB. It was found that GP with radial basis kernel showed the highest performance with an accuracy of 81.5%.

Yet another predictive model for the diagnosis of diabetes using the PID dataset was proposed by Sisodia *et al.* [25] wherein no data preprocessing was employed. Three ML classifiers including the SVM, DT, and NB were employed and the highest accuracy of 76.3% was achieved through the NB classifier.

Another recent research by Wang *et al.* [26] proposed a diabetes prediction algorithm on the PID dataset using the techniques for filling the missing values and reducing the influence of class imbalance on the prediction. The former is achieved through NB method for normalizing the data and the latter is achieved through an Adaptive Synthetic sampling method (ADASYN). Finally, the classification is done through the RF classifier. The performance evaluation is done through K-fold Cross Validation and the data preprocessing does not include any feature selection techniques. The method is compared with other classifiers including the NB, SVM, DT and RF (with no data preprocessing). The proposed algorithm achieved the highest accuracy of 87.1% with $K = 5$ in K-fold CV.

A recent research by Islam *et al.* [8] which dealt with early stage diagnosis of the risk of diabetes employed a novel dataset that comprised the data about recent diabetics or prospective diabetics. The dataset was collected through patients at Sylhet Diabetes Hospital in Sylhet, Bangladesh. The ML classifiers including the NB, LR, and RF were employed. The performance evaluation was done through the techniques of train-test split as well as K-Fold Cross Validation. The highest performance was achieved with RF classifier having an accuracy of 97.4% with K-Fold Cross validation and 99% accuracy with train-test split.

B. PREDICTIVE MODELING OF DIABETES WITH FEATURE SELECTION

Most of the ML frameworks for diabetes prediction strive to maximize the classification accuracy with minimal features by discarding the irrelevant features.

A method employing the feature selection along with ensemble technique is proposed by Dewangan *et al.* [27] where an ensemble of multi-layer perceptron and Bayesian net classifiers is used along with information gain feature selection technique. An accuracy of 81.89% was achieved with the reduced feature subset of 6 features out of the full feature set of 8 features in the PID dataset.

A ML framework named as Hierarchical Multi-level classifiers Bagging with Multi-objective optimized voting (HM-BagMoov) involving a hierarchy of heterogeneous classifiers was proposed by Bashir *et al.* [28] which involved bagging and optimized weighting. The framework employed data preprocessing steps which included noise removal, outlier detection, missing value imputation and feature selection. The F-score method is employed for feature selection to choose the most relevant features along with a threshold. Each feature has its F-score computed and the average of the F-score of all features decides the threshold. All features with F-score less than the threshold are discarded. Three layered ensemble approach is used wherein the first layer involves NB, Linear Regression, QDA, Instance Based Learner (IBL), and SVM. The layer-1 output is fed into layer-2 through weighted bagging ensemble using two classifiers of ANN and RF. Finally, in layer-3 the final prediction is done through multi-layer weighted bagging ensemble approach.

An attempt to improve the prediction accuracy of diabetes diagnosis approaches through ML algorithms was done by Vaishali *et al.* [29]. The PID dataset was used and the feature selection was performed through Goldberg's Genetic algorithm by Goldberg *et al.* [30] and the classification was done through an evolutionary approach employing a multi-objective fuzzy classifier. The working principle of the fuzzy classifier is to achieve a maximum accuracy with minimum rules. The outcome was a reduced feature subset which is half the size of original feature set. The train-test split (70%-30%) method was used for performance evaluation and an accuracy of 83% was achieved.

A classification framework aimed to achieve highest risk stratification accuracy for diabetes was proposed by Maniruzzaman *et al.* [31] wherein the PID dataset was initially subjected to data preprocessing techniques including the missing values and outliers replacement with median followed by feature selection through techniques like RF, mutual information by Peng *et al.* [32], LR, Analysis Of Variance(ANOVA) by Howell [33], Principal Component Analysis(PCA) by Rao [34], and Fisher discriminant ratio. This was followed by the usage of various classifiers like LDA, QDA, NB, GPC, SVM, ANN, AB, LR, DT, and RF. It was found that when RF algorithm is used for feature selection as well as classification, highest accuracy of 92.26% was achieved.

A recent research work by Kaur *et al.* [35] dealt with developing ML models for the detection of diabetes using the PID dataset through the algorithms of SVM with Linear Kernel, SVM with Radial Basis Function kernel, KNN, ANN and Multifactor dimensionality reduction. They also employed the wrapper feature selection algorithm called as Boruta. The performance evaluation of the models showed that highest accuracy of 89% was achieved with SVM linear model.

Another recent work by Li *et al.* [36] on PID dataset uses filter based feature selection method called as Coefficient of Variation (CV) wherein the features with low dispersion are eliminated from the model. After the filtering of irrelevant features, two classifiers of DT and Multilayer Perceptron are used with 10-fold cross validation. With the size of the feature subset reduced to 4, an accuracy of 77% was achieved with multilayer perceptron model.

Another recent work on the early detection of Diabetes Mellitus was proposed in [37] wherein the accuracy was improved through a correlation based feature selection method. The classifiers employed were NB, DT, RF and SVM. The original feature set of 15 features was reduced to 11 features and the highest accuracy of 82.3% was achieved with Naive Bayes' classifier.

A robust framework employing various data preprocessing techniques for data cleaning and feature selection along with the usage of various ML classifiers and their ensembles was proposed by Hasan *et al.* [38]. The data preprocessing involved outlier rejection, filling the missing values, data standardization. The feature selection was done through PCA, Independent Component Analysis (ICA) by Hyvärinen *et al.* [39], and Correlation-based technique by Han *et al.* [40]. The classifiers used were KNN, DT, RF, AB, NB, and XGBoost by Chen *et al.* [41] and Multilayer Perceptron (MLP). The ensemble technique uses weights for each of the models' predictions computed based on Area Under Receiver Operating Characteristic (ROC) Curve (AUC) of the ML model. The best performing ensemble with a highest AUC of 0.95 was found to be the combination of AB and XGBoost along with the preprocessing steps of outlier rejection and filling of missing values.

C. LIMITATIONS OF EXISTING METHODS

All of the existing methods involve ML models that achieved a maximum accuracy within 95% and most of them employed the PID diabetes dataset which is restricted to female patients with mere 8 attributes. Hence the dataset is less representative. The two main drawbacks of the existing literature including the lower prediction accuracy as well as limited representativeness of the input dataset are overcome in the proposed research as follows:

The current research work utilizes the early stage diabetes diagnosis dataset by Islam *et al.* [8] available at UCI machine learning repository. The dataset has 520 instances and 16 attributes which contribute to the representativeness of the dataset. The research work achieves highest accuracy with reduced features and classification done through a wide range of heterogeneous classifiers applied individually and also in the form a stacking ensemble to maximize the accuracy.

III. MATERIALS AND METHODS

In this section, we present the description of the dataset used and the different components within the proposed predictive model representing the terms materials and methods respectively. Sections III A presents the dataset description along with the distribution of positive and negative classes across all attributes. Section III B presents the details of the proposed model along with various preprocessing mechanisms and the cross validation scheme. Finally, section III C presents the details of the hardware platform and performance evaluation metrics.

A. DETAILS OF THE DATASET EMPLOYED

In this research, the publicly available dataset in the UCI repository for Early diabetes diagnosis [8] was used. It consists of data related to 16 different attributes collected through a survey regarding their physical conditions. This simple information about the various binary valued attributes can be used for the timely diagnosis of the onset of diabetes. The dataset contains diabetes specific symptoms collected from 520 patients. Out of the 520 instances, 320 fall under the category of positive cases whereas 200 fall under negative cases. It can be observed that the dataset is not unbalanced as the imbalance ratio is 200:320 which is 5:8. Hence the performance metrics employed are precision, recall, F-Measure, Accuracy and Area under ROC Curve as these are more suitable for balanced classification.

The data generation was done through a direct questionnaire given to people who were diagnosed with diabetes recently as well as people who have not yet been diagnosed despite one or more symptoms. The process of data generation was done at Sylhet Diabetes Hospital, Bangladesh. The description of attributes and the related statistical information are provided below. The distribution of positive and negative classes across all the 16 attributes is shown in Fig. 1 The dataset employed in the proposed research concerns type 2 diabetes which intends to predict the likelihood of diabetes risk at its early stage, using different symptoms of the patients. As type 2 diabetes is an adult-onset diabetes and it is very common in adults with obesity, prediabetes and weakness etc., the dataset includes data related to such symptoms that may cause diabetes.

Each feature of the dataset contributes to a symptom linked to the onset of diabetes. As can be seen from the attribute description in Table 1, out of 16 features, there is 1 numeric attribute (Age) while the remaining 15 attributes are categorical. Hence, we employed one-hot encoding to convert the categorical attributes to numeric ones. After the one-hot encoding, the 15 categorical features are transformed into 30 numeric features ($15 \times 2 = 30$) since each feature can have only two values.

The feature engineering phase employed in the current research involves various techniques using a combination of feature extraction and feature selection. The feature selection employs ANOVA filter and Crow Search algorithm (CSA) [42] whereas the feature extraction employs the Singular Value Decomposition technique (SVD) [43]. The feature engineering pipelines involve various combinations of ANOVA, CSA, and SVD techniques corresponding to each of the experiments. The outcome of preprocessing the dataset

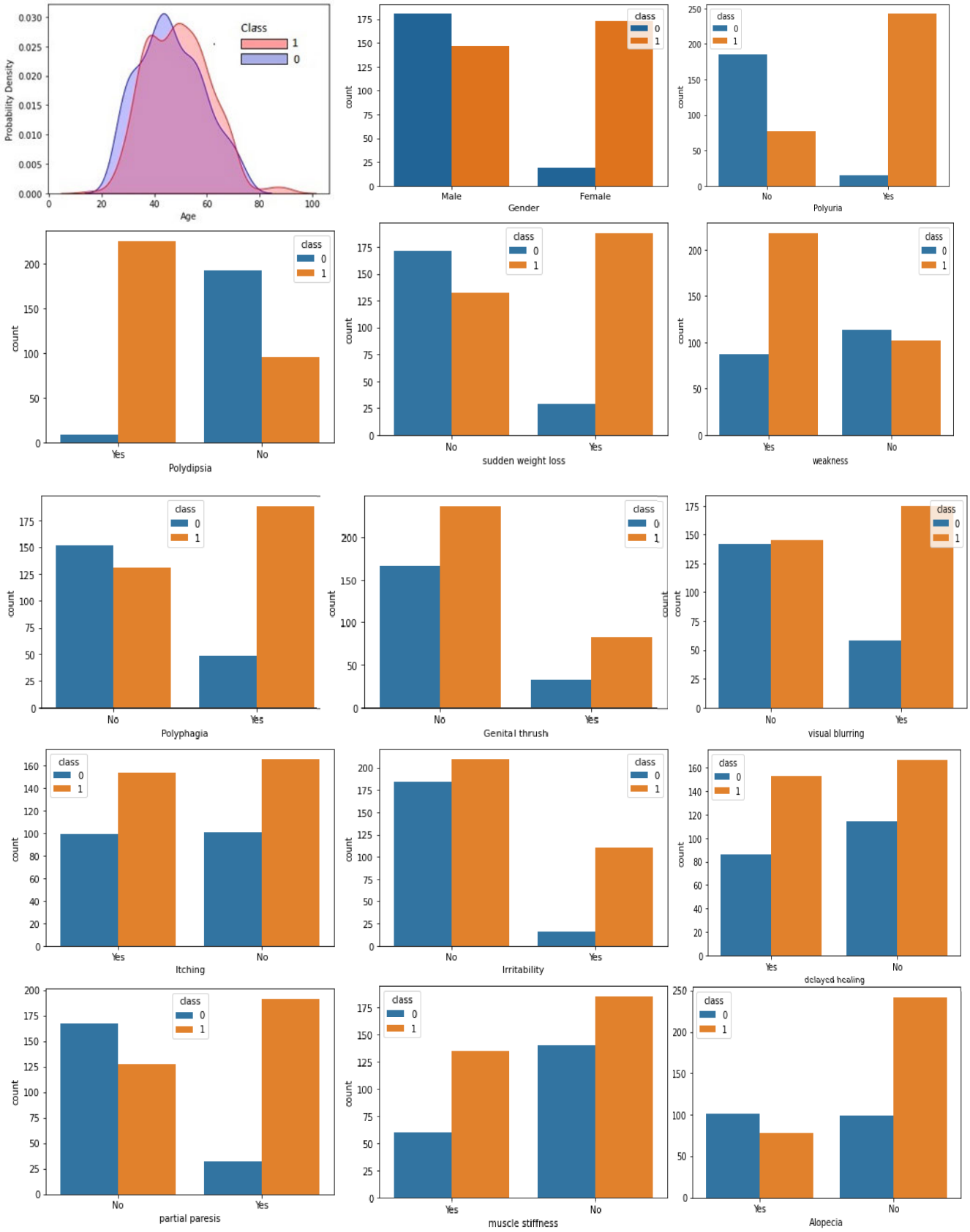

FIGURE 1. Class Distribution across different attributes in the dataset.

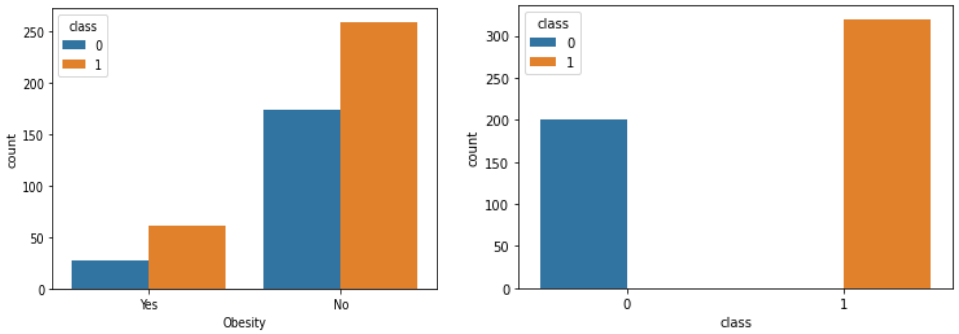

FIGURE 1. *(Continued.)* **Class Distribution across different attributes in the dataset.**

TABLE 1. Dataset description.

S. No	Attributes	Values	Type of Attribute
1	Age	>=20 and <=90	Numeric
2	Sex	1.Male, 2.Female	Categorical
3	Polyuria	1.Yes, 2.No.	Categorical
4	Polydipsia	1.Yes, 2.No.	Categorical
5	Sudden Weight Loss	1.Yes, 2.No.	Categorical
6	Weakness	1.Yes, 2.No.	Categorical
7	Polyphagia	1.Yes, 2.No.	Categorical
8	Genital Thrush	1.Yes, 2.No.	Categorical
9	Visual Blurring	1.Yes, 2.No.	Categorical
10	Itching	1.Yes, 2.No.	Categorical
11	Irritability	1.Yes, 2.No.	Categorical
12	Delayed Healing	1.Yes, 2.No.	Categorical
13	Partial Paresis	1.Yes, 2.No.	Categorical
14	Muscle Stiffness	1.Yes, 2.No.	Categorical
15	Alopecia	1.Yes, 2.No.	Categorical
16	Obesity	1.Yes, 2.No.	Categorical
17	Class	1.Positive, 2.Negative	Categorical

with each of the feature engineering pipelines is a reduced feature subset.

B. PROPOSED PREDICTIVE MODEL

The proposed predictive model is illustrated through Fig. 2 and the dataflow can be explained as follows: The data preparation is the crucial step which involves various data transformations and feature engineering methods to form various feature engineering pipelines. The data transformations include the one-hot encoding followed by data standardization through standard scaling transform. This is followed by dimensionality reduction through feature selection and feature extraction techniques to generate a more representative and reduced feature set. The reduced feature set is subjected to the process of generating training, validation and test datasets through a combination of train-test split and repeated stratified K-fold cross validation. Multiple heterogeneous classifiers along with their stacking ensemble are used for training and the best model is selected using validation dataset. The selected best model is evaluated using the test dataset followed by reporting of evaluation metrics.

1) ONE HOT ENCODING

Most of the machine learning algorithms require numeric data in all the input and output variables and hence the categorical data has to be converted to numeric one. When there is no ordinal relationship between the categorical data, the one-hot encoding is the most appropriate.

2) STANDARDIZATION

As the performance of a machine learning algorithm is negatively affected when the attributes within the dataset have different scales, data standardization becomes important. It is the process of rescaling the distribution of values such that the mean has the value zero and standard deviation has the value 1.

In the diabetic dataset used in the current research, the age attribute is a numerical with a skewed distribution and varying scale as compared to remaining other attributes generated through one-hot encoding. Hence we employ the standard scaling transform in the data preparation.

3) FEATURE SELECTION

It is a method of decreasing the feature space by selecting most relevant input attributes that have the strongest relationship with the target attribute. Having less features not only decreases the computational and storage cost of the predictive model but also increases the performance. A number of advantages are associated with reducing the size of the feature set. For instance, the overfitting of data can be reduced since the elimination of irrelevant features implies reduction of noise. The accuracy of model also increases since misleading input variables has been removed. Also, the reduction in the size of the data implies lesser time in the construction and training of the model.

The conventional feature selection methods used in supervised learning come into the three categories of filter, wrapper and intrinsic methods. In the current work, we employ two different types of feature selection: filter based method and wrapper method. The feature engineering pipelines are built using various combinations of these methods with optimal number of features along with SVD for feature extraction.

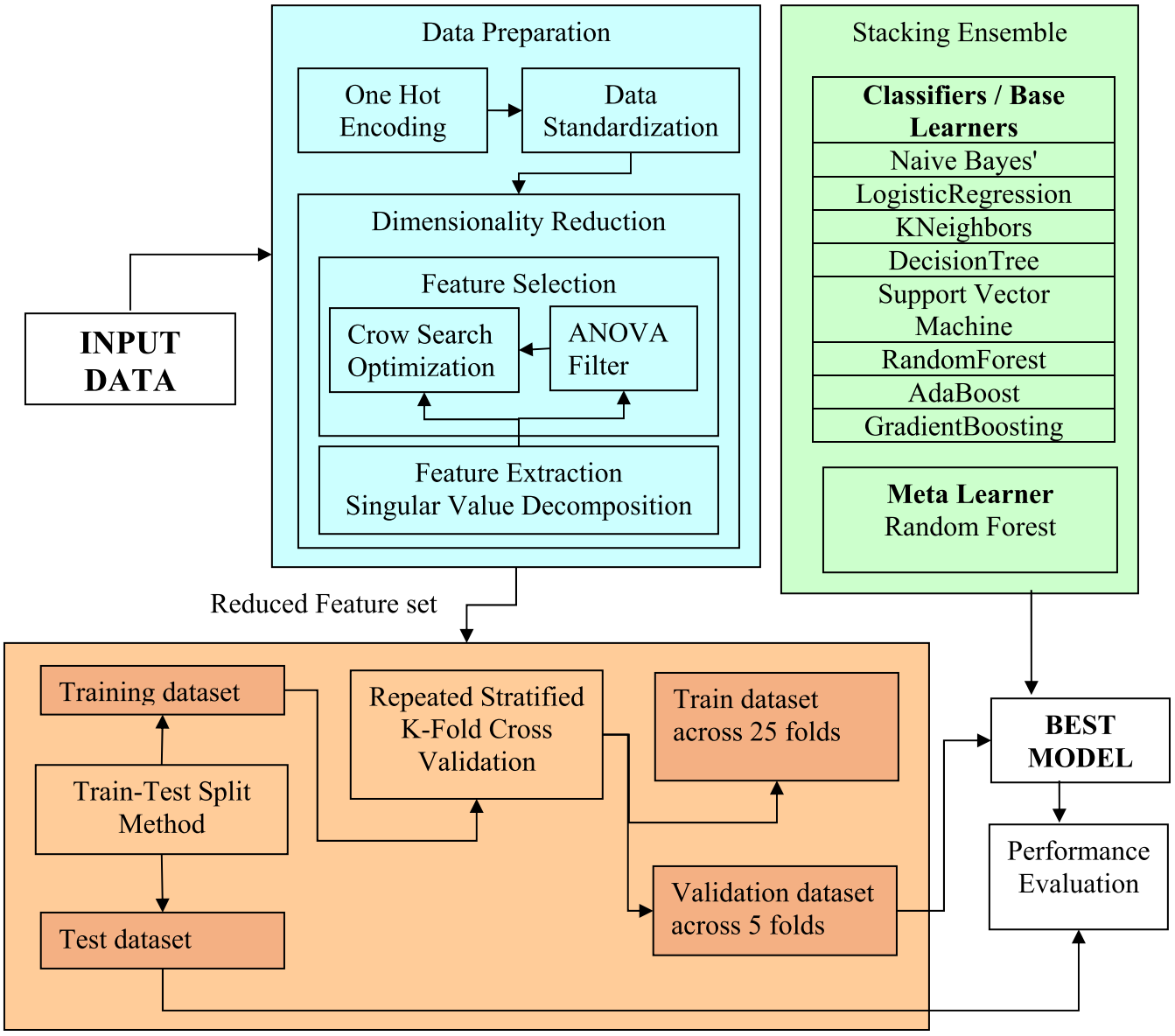

FIGURE 2. Block diagram of the proposed predictive model.

a: FILTER BASED METHOD

Filter-based methods are based upon the evaluation of the statistical relationships between the input features and the target variable. The input features having a strong relationship with the target variable are selected and others are discarded. Each feature receives a score representing the correlation with the target variable using which filtering is done to select the most relevant features.

The statistical measures employed for relationship evaluation are based upon whether the types of input and variables are numerical or categorical. As the dataset employed in this research has 31 numerical features after on-hot encoding and the target variable is categorical, we employ the anova-f value as a statistical measure.

b: WRAPPER METHOD

Wrapper methods use different subsets of feature set and evaluate the performance of an ML model for each subset. A performance evaluation score is assigned to each subset and the one with the highest score is selected. The search for the optimal feature subset can be methodical or stochastic. The former approach may employ a strategy like best-first search and the latter one may employ an algorithm like random hill-climbing.

Specifically, the wrapper method used in the proposed framework is a bio-inspired metaheuristic method called a crow search method. As we know, feature selection is the process of selecting those features which maximize the accuracy of an ML model, it can be treated as an optimization problem, especially when the feature set size is large. The optimization

function has got to minimize the size of feature subset and the classification error. Assuming that, there are N features, the total number of possible subsets are 2^N. The search space size increases exponentially and it is impractical to employ a brute-force approach using an exhaustive search. Hence, meta-heuristics are used to reduce the exploration of search space. In this context, a number of algorithms have been proposed which include Particle Swarm Optimization [44] [45], Artificial Bee Colony [46], Differential Evolution [47], Bat algorithm [48], Moth Flame.

Optimization [49], Grey Wolf Optimizer [50], Bufferfly optimization algorithm [51], Sine Cosine Algorithm [52] and CSA. The CSA is a recently proposed algorithm which has gained great attention from the research community attributed to its simplicity, ease of implementation, efficiency, and fewer parameters.

c: CROW SEARCH ALGORITHM

The crow search algorithm is one of the recently proposed algorithms which is bio-inspired optimization algorithms based upon the intelligent behavior of crows within their flock during the food collection. The approach has been proved to outperform many well-known algorithms like genetic algorithm and particle swarm optimization.

In the current research, the application of CSA is done towards optimal feature subset selection for obtaining the highest accuracy with minimal features. The implementation of the feature selection function utilizes the baseline version of CSA and its enhanced version termed as Enhanced CSA (ECSA) [53] for feature selection.

Specifically, the process starts by the initialization sub-routine which determines the initial positions for the set of N crows. The position of each crow at any point of time represents a solution vector and the best solution has to be determined based upon a fitness function.

The working of CSA resembles the behavioral traits of a crow in tapping the food resources of its peers in the flock. There is always an ongoing survival battle within the flock where each crow not only attempts to safeguard its own hideout location but tail other crows for plundering their food. There are two situations in this scenario which can arise based upon whether the followed-up crow is aware that it is being followed. If it is aware, then it will misdirect the follower to a false hideout spot. Otherwise, it becomes a victim and loses its food since its hideout spot is revealed to the follower.

The standard CSA has the entire flock distributed across the search space of d-dimensional decision area. The position of each crow in search space is an array of decision variables which has to fall within the boundary of decision area. The algorithm works iteratively where each crow attempts to secure its own hide-out spot and find others hide-out spots. The position of a crow i at iteration t is represented by x_i^t and the best hide-out spot so far is represented by the memory of the crow as m_i^t. The crow i randomly selects another random crow j to follow. The entire process is repeated iteratively for a predetermined number of iterations to achieve the best possible solution. This process of crow i following other peer j is done for all crows i such that $i = 1, 2, 3 \ldots N$, $j = 1..N$ and j is not equal to i.

The update to the position of crow i depends upon the awareness of crow j and it is represented as in (1):

$$x_i^{t+1} = \begin{cases} x_i^t + r_i \times fl_i^t \times \left(m_j^t - x_i^t\right), & r_i \geq AP_j^t \\ a\ random\ position, & otherwise \end{cases} \quad (1)$$

where fl_i^t is the flight length of crow i at iteration t and awareness probability of crow j at iteration t is represented by and r_i is the random number with uniform distribution in the range [0,1].

The update of the memory of crow i based upon its new position is done depending upon the fitness function evaluated upon the new position. If the existing memory's fitness is lesser than the newly discovered position, then the memory is updated with the new position, otherwise it remains the same. Specifically, it is given as in (2):

$$m_i^{t+1} = \begin{cases} x_i^{t+1}, & fitness(x_i^{t+1}) > fitness(m_i^t) \\ m_i^t, & \text{otherwise} \end{cases} \quad (2)$$

The application of CSA to the problem of feature selection is done as follows: Initially, the lower and upper boundaries of the search space are chosen as l and u respectively (which are predetermined values). Then the initial positions are computed as in (3):

$$x_{i,j} = l - (l - u) * r \quad \text{i} = 1 \ldots \text{N}, \quad \text{and j} = 1 \ldots \text{d} \quad (3)$$

where r ϵ [0.1] is a random value and N is the number of crows and d is number of features/dimensions of the decision space. The position of each crow is binarized using the sigmoid function as in (4):

$$x_{i,j}^t = \begin{cases} 1, & \text{Sigma}(x_i^t) \geq \sigma \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $x_{i,j}^t$ is the continuous-valued input vector x_i at dimension j and $\sigma \epsilon$ [0.1] is a random value which represents the threshold. The Sigma(k) function is logistic sigmoid function defined as in (5):

$$Sigma(k) = \frac{1}{1 + e^{-k}} \quad (5)$$

The solution vector corresponds to the position of each crow in the solution space and the number of dimensions in the decision space corresponds to the number of features selected within the feature subset. The best solution is evaluated using the fitness function which is an optimization function to be minimized. It is a combination of classification accuracy and the number of features selected. As it is a minimization function, the intent is to minimize the classification error and the number of features in the feature subset. After binarization, each solution vector comprises a sequence of 1 and 0s with the fixed length d. The positions with a 1 represent the features to be included from the original feature set to form an optimal reduced size feature subset

to be used for classification with the fitness function. The ML algorithm used for classification through the selected features within the fitness function is KNN. Mathematically, the fitness function is represented as in (6):

$$fitness(x_i^t) = \alpha \times Err(x_i^t) + (1 - \alpha) \times \frac{|S|}{|F|} \tag{6}$$

where $Err(x_i^t)$ is the classification error upon the using the reduced feature subset S obtained by applying the solution vector x_i^t upon the given feature set F of the input dataset. $|S|$ is the size of subset S and $|F|$ is the size of set F, α is a parameter in [0,1] which represents the weight given to error and $(1 - \alpha)$ represents the weight given to feature set reduction.

In our implementation, we have used α value as 0.9 to specify that the classification accuracy is given 90% weight and feature reduction is given 10% weight. The number of crows is fixed to 10 and the number of dimensions in the decision space is 31 which is the number of features after one-hot encoding is applied to the original dataset.

Using, the above parameters, the algorithm comprises fixed number of iterations, represented by t_{max} (which is set to 100). In each iteration, the process of each crow following randomly chosen other peer crow is repeated for all the crows computing the fitness value for each crow. At the end of each iteration, the crow with lowest fitness (classification error and feature subset size) is saved as the optimal solution for that iteration. After all the iterations, the optimal solutions across all iterations are considered and the one with lowest fitness (classification error and feature subset size) is selected as the best solution.

The following algorithm gives the insights about the application of CSA in the context of feature selection:

Crow Search Algorithm (CSA) for Feature Selection

Parameter Initialization

T_{max} (maximum number of iterations) = 30
Pd (Problem dimension or number of features) = 31
N (Flock size) = 10
AP (Awareness probability) = 0.1
FL (Flight length) = 2

1. Initialize randomly positions of each crow X_i (1,2. . . Pd) for i = 1,2, . . . N using Eq. (3) followed by binarization through Eq. (4) and Eq. (5).
2. Initialize the memory of each crow M_i (1, 2, ...pd) for i = 1,2..N using the positions computed in step 1.
3. Evaluate the position of each crow using the fitness function given in Eq (6).
4. For $t = 1$: T_{max}
5. For $i = 1$: N
6. Randomly select one crow to follow
7. Update crow's position using Eq. (1) followed by binarization through Eq. (4) and Eq. (5)
8. End For
9. Check feasibility of new positions according to problem constrains
10. Assess the updated position of the crows through the fitness function given in Eq. (6)
11. Update the memory of crow M_i(1, 2, . . . pd) as per its new position using Eq. (2)
12. $t = t + 1$
13. End For
14. Return the best solution Mi consisting of selected features such that fitness(M_i) = max (fitness (M_1, M_2, . . . M_N))

4) FEATURE EXTRACTION

It is a method of constructing new features using the given set of original features. It is basically a data preparation step which creates a data transform such that low dimensional data is created from the given higher dimension data in a way that preserves the important relationships within the data. It differs from feature selection in that, it creates an entirely new set of features which are incomparable with the original feature set whereas feature selection relies upon creating a subset of the most relevant features from the given feature set.

Since there are 15 features which are one-hot encoded (out of a total of 16), the resultant dataset is a sparse dataset. Hence the SVD technique from linear algebra is used to create a projection of sparse dataset before fitting the model. It creates a projection of a sparse dataset with reduced dimensions/features using which a machine learning model can be constructed. The projection approach to feature reduction results in a decrease in the dimensions ensuring that structural relationships among the variables within the given dataset are preserved.

5) REPEATED STRATIFIED K FOLD CROSS VALIDATION

Cross Validation(CV) is a technique which uses resampling to evaluate the performance of a machine learning algorithm. It is associated with a single parameter K that represents the number of folds or partitions of the dataset, out of which $K-1$ partitions are used for training and 1 partition is used for testing/validation. This process is repeated for each partition until each one of them becomes a test set. For each iteration, the accuracy/performance estimates are stored and the final estimate across all iterations is computed as in (7):

$$P = \frac{1}{k} \times \sum_{i=1}^{k} P_i \pm \sqrt{\frac{\sum_{i=1}^{k} (P_i - \bar{P})^2}{k - 1}} \tag{7}$$

where P is the final performance metric, k is the number of folds, P_i is the performance metric for the i^{th} fold, $\bar{P}$ is the mean of the performance metric across all folds.

Usually, the CV method is used to determine the performance of an ML model on unseen data. This is done by

limiting the sample size and estimating the generalized performance on the data not used in training. It is more popular than its counterpart like train/test split method since it has less bias and generates a lower optimistic estimate of the model's performance.

Even though K-fold cross validation is the standard approach used for performance evaluation of ML model, a single iteration of the K-fold CV might result in a noisy performance estimate since varying the splits/samples of the dataset may generate varying results. This problem can be overcome through Repeated Stratified K-fold CV wherein the CV process is repeated for a specified number of iterations such that the generation of samples for different folds ensures that same amount of instances with a given class label appear across all folds.

The diabetes dataset is first split into training dataset and test dataset using train-test split method using the split percentage as follows: train 80% and test 20%. The training dataset termed as 'train_val' is subjected to repeated stratified K-fold as the volume of the data is relatively low. Within 'train_val' dataset, the first 25 folds are used for training and the remaining 5 folds are used for validation to choose the best model by computing robust performance estimates. Following this, the tuned model is used upon the test dataset to test the selected model and report the performance estimates.

The data leakage is prevented by employing a strategy of holding back a validation dataset for the final sanity check of the developed models. That is done by splitting the training dataset into train and validation sets, and store away the validation dataset. The performance evaluation of the models is done using the validation dataset which aids in selecting the best model. Testing of the selected model is done using the test dataset (different from training and validation datasets) and the evaluation metrics are reported.

The approach is more suitable for small to moderately sized datasets due to its computational complexity. Hence it is the most appropriate method which can be employed with diabetes dataset used in this paper.

6) ML MODELS AND STACKING ENSEMBLE

It is a special type of ensemble machine learning algorithm which works by combining the predictions of multiple heterogeneous models working at two different hierarchical levels on the same dataset.

Usually, for a given dataset different models may be effective in different ways. The stacking ensemble attempts to amalgamate the predictions of typically different ML models to achieve highest accuracy than the individual models.

Even though, there are well known approaches like bagging and boosting used for creating ensembles, the stacking ensemble has a distinct approach as it employs different ML models of varying kind. It employs a single model termed as meta model to interpret the best combination of predictions from the base models. The stacking ensemble has an architecture comprising two levels known as level 0 and level 1. The former involves heterogeneous ML models termed as base learners and the latter involves a single model called as meta learner whose role is to perform the unification of base learners' predictions.

The level 0 models are trained on the entire dataset and the training of the level 1 model is based upon predictions of level 0 models upon out-of-sample data.

In other words, the level 0 models are fed with the data not used to train them. The predictions made by them along with the actual outputs are used as input and output pairs respectively of the training data used to fit the level 1 model. In case of classification tasks, the outputs from the level 0 models which act as input to level 1 model are probability values or class labels.

The preparation of training data for the level 1/meta model is the use of k-fold cross validation upon the base models and the use of out-of-fold predictions as the training data of the level 1 model. The input fields of the base models' training data can also be incorporated within the training data of the meta model. It helps the meta model to learn the best way of combining the base learners. Stacking ensemble is most appropriate when the prediction errors of different base models are not highly correlated. The selected base learners can be quite complex as well as diverse. The best performance is usually achieved with models which have varying assumptions about the predictive model and use different internal representations of data like trees and samples. It is also possible to use other ensemble algorithms like Random Forest, Gradient Boosting, Adaboost as the base learners. For the meta model, it is a usual practice to use simple linear model like logistic regression for unifying the base learners' predictions. But the best choice of the meta model is often dependent upon the specific dataset and the attributes' statistical relationship with the target variable.

Even though, stacking ensemble's intent is the improvement in accuracy, it might not always be possible as it is dependent upon the specific dataset's complexity. Another factor which affects the performance is representational capability of training data which enables further learning by unifying the predictions of base learners.

Owing to these factors, it is definitely possible that base learners performance may be equal/greater than the stacking ensemble. Under these circumstances, it is best to employ the base models since they involve less computational complexity.

In the current research work, we have employed NB, KNN, LR, DT, SVM, Gradient Boost, AdaBoost, and RF classifiers as base learners and the meta learner is chosen to be RF Classifier. The selection of base learners is done empirically through extensive experimentation owing to the fact that a stacking ensemble requires the use of a diverse range of models that make varying assumptions about the predictive modeling task such as linear models, decision trees, support vector machines, and other ensemble algorithms such as random forests. Through experimentation, it was found that random forests were giving highest accuracy compared

TABLE 2. Time complexity of feature engineering pipelines.

Feature Engineering Approach	Time Complexity Computation	Time Complexity
None	----------	----------
CS	tNd * (nd) = $O(nd^2)$	$O(nd^2)$
Filter	O(nd)	O(nd)
Filter(25)+CS	O(nd)+$O(nd^2)$ = $O(nd^2)$	$O(nd^2)$
SVD (13)	$O(d^3+nd^2)$	$O(d^3+nd^2)$
SVD(20)+ CS	$O(d^3+nd^2)$+$O(nd^2)$ = $O(d^3+nd^2)$	$O(d^3+nd^2)$
SVD(13)+CS	$O(d^3+nd^2)$+ $O(nd^2)$ = $O(d^3+nd^2)$	$O(d^3+nd^2)$
SVD(9)	$O(d^3+nd^2)$	$O(d^3+nd^2)$
SVD(13)+ Filter(8)	$O(d^3+nd^2)$ +O(nd) = $O(d^3+nd^2)$	$O(d^3+nd^2)$
SVD(8)	$O(d^3+nd^2)$	$O(d^3+nd^2)$
SVD(13)+ Filter(10)+CS	$O(d^3+nd^2)$ + O(nd) + $O(nd^2)$ = $O(d^3+nd^2)$	$O(d^3+nd^2)$

TABLE 3. Time complexity of ML classifiers.

Name of the Classifier	Time Complexity
NB	O(nd)
KNN	O(nd)
LR	O(nd)
SVM	$O(n^3)$
DT	O(nd log n)
RF	O(knd log n)
AB	O(knd log n)
GB	O(knd log n)
Stack	$O(kn^4d \log n)$

k : number of decision trees

to other base learners. Based on this, random forest is used as meta learner.

7) PSEUDOCODE OF THE PROPOSED OPTIMAL MACHINE LEARNING PIPELINE

Notations:

PEP: Preprocessing Pipeline.
FEPList: Feature Engineering Pipeline List.
SE: Stacking Ensemble.
CLPList: Classifier Pipeline List.
FET: Feature Engineering Technique.
CLT: Classification Technique.
RSKF: RepeatedStratifiedKFold.
FEP_DEF: Definition of Feature Engineering Pipeline structure comprising the size of reduced feature subset, highest accuracy with one of the classifier in CLPList and the full machine learning pipeline employed.
EXP_DEF: Definition of the individual experiment structure comprising the size of reduced feature subset, accuracy with the chosen classifier in the CLPList and the full machine learning pipeline employed.

Method:

Input: Diabetes Dataset
Output: Minimum feature Pipeline and Maximum accuracy Pipeline

1. Read Input Dataset
2. Transform Categorical features to numeric ones through One-hot Encoding
3. Input Dataset -> OneHotEncoding (Input Dataset)
4. PEP -> StandardScaler
5. FEPList <- {CSA, ANOVA, SVD, ANOVA ANOVA (25)+CSA, SVD(13), SVD(20)+CSA, SVD(13)+CSA, SVD(9), SVD(13)+ANOVA, ANOVA(8), SVD(8), SVD(13)+ANOVA ANOVA(10)+CSA }
6. SE -> {BaseLearners (NB, KNN, LR, SVM, DT, RF, AB, GB), Meta Learner(RF)}
7. CLPList -> {NB, KNN, LR, SVM, DT, RF, AB, GB, SE }
8. DEFINITION FEP_DEF: {FeatureSubsetSize, Accuracy, MLP}
9. DEFINITION EXP_DEF: {FeatureSubsetSize, Accuracy, MLP}
10. DECLARE FEP: Array [1..FEPList.Length] of FEP_DEF
11. DECLARE EXP: Array [1..FEPList.Length x CLPList.Length] of EXP_DEF
12. i <- 1
13. k <- 1
14. For each FET in FEPList do
15. FEP(i).Accuracy <- 0
16. For each CLT in CLPList do
17. EXP(k).MLP<- (PEP, FET, CLP)
18. Train-Test-Split (Dataset)-> (Train_Val, Test)
19. RSKF(repeats=3, Folds=10, Train_Val)-> (Train(25 folds), Validation(5 folds))
20. Validation(5 folds)-> MLP
21. EXP(k).Accuracy <- CrossValScore (EXP(k). MLP, Test)
22. EXP(k).FeatureSubetSize-> FEP(i).FeatureSubsetSize
23. If (EXP(k).Accuracy > FEP(i).Accuracy) then
24. FEP(i).Accuracy->EXP(k). Accuracy
25. FEP(i).MLP-> EXP(k).MLP
26. EndIf
27. k <- k +1
28. EndFor
29. i <- i + 1
30. EndFor
31. FEP_MaxAccuracy -> Max(FEP(i).Accuracy)
32. FEP_MinFeatures -> Min(FEP(i).FeatureSubsetSize)
33. For i = 1….FEPList.Length do
34. If (FEP(i).Accuracy == FEP_MaxAccuarcy) then
35. Print "Max Accu Pipeline:" +FEP(i). MLP
36. EndIf
37. If FEP(i).FeatureSubsetSize == FEP_MinFeatures then
38. Print "Min Featu Pipeline:" + FEP(i). MLP
39. EndIf
40. EndFor

8) TIME COMPLEXITY

The computation of time complexity of the above pseudo-code requires us to consider the time complexity of the individual feature engineering pipelines and

TABLE 4. Time complexity of ML pipelines.

Machine Learning Pipeline	Time Complexity
None + CLPList	$O(kn^4dlogn)$
CS + CLPList	$O(nd^2) + O(kn^4dlogn)$
Filter + CLPList	$O(nd) + O(kn^4dlogn)$
Filter(25)+CS + CLPList	$O(nd^2) + O(kn^4dlogn)$
SVD (13) + CLPList	$O(d^3+nd^2) + O(kn^4dlogn)$
SVD(20)+ CS + CLPList	$O(d^3+nd^2) + O(kn^4dlogn)$
SVD(13)+CS + CLPList	$O(d^3+nd^2) + O(kn^4dlogn)$
SVD(9) + CLPList	$O(d^3+nd^2) + O(kn^4dlogn)$
SVD(13)+ Filter(8) + CLPList	$O(d^3+nd^2) + O(kn^4dlogn)$
SVD(8) + CLPList	$O(d^3+nd^2) + O(kn^4dlogn)$

CLPList: { NB, KNN, LR, SVM, DT, RF, AB, GB, Stack}

the individual ML classifiers employed for classification. Tables 2, 3, and 4 give the time complexity of the individual feature engineering pipelines, classifiers and the ML pipelines employing the different combinations of feature engineering pipelines and the classifiers respectively. In tables 2, 3, and 4, the symbol n represents the number of records in the dataset (rows) and the symbol d represents the number of dimensions/features (columns).

For the feature engineering pipelines involving ANOVA filter and SVD, the time complexity is given in table 2 [58]. For the Crow search algorithm employed for feature selection, the time complexity computation is as follows:

$$t \times N \times d \times (n \times d) = O(nd^2)$$

where t is the number of iterations and N is the number of crows.

For the standard classifiers, NB, KNN, LR, SVM, DT, RF, AB, and GB the time complexity is given in table 3 [59] and [60]. For the stacking ensemble classifier, the time complexity can be computed by considering time complexities of meta-classifier and base-classifiers as follows:

Meta-Classifier * (Base-Classifier$_1$+ Base-Classifier$_2$ + ... + Base-Classifier$_n$) = RF*(NB + KNN + LR + SVM+ DT + RF + AB + GB) = knd log n * (nd + nd + nd + n^3+ nd log n + knd log n + knd log n + knd log n) = O(knd log n * (n^3+ nd + knd log n)) = O(kn^4d log n)

For the entire pseudo-code, computational complexity can be obtained by considering the sum of the computational complexities of all the pipelines as follows:

$$O(d^3 + nd^2) + O(kn^4dlogn)$$

9) EXECUTION TIME

The execution time of the different ML pipelines is given in table 5.

When the feature engineering pipelines are considered individually (without any combination), it can be observed that the feature engineering pipeline with CS has the highest execution time due to its inherent complexity and the number of iterations for feature selection. SVD has the second highest execution time followed by ANOVA filter.

TABLE 5. Execution time of ML pipelines.

ML Pipeline	Execution Time in Seconds
None + CLPList	116
CS + CLPList	174
Filter + CLPList	99
Filter(25)+CS + CLPList	158
SVD (13) + CLPList	128
SVD(20)+ CS + CLPList	180
SVD(13)+CS + CLPList	187
SVD(9) + CLPList	133
SVD(13)+ Filter(8) + CLPList	131
SVD(8) + CLPList	125
SVD(13)+Filter(10) + CS + CLPList	157

C. EVALUATION METRICS

The different ML models using the various proposed data preprocessing pipelines are implemented using the Python Programming language through the Anaconda Distribution running on a machine with Windows 10 Operating System and the hardware configuration as follows: Intel(R) CoreTM i5-3210M CPU @ 2:50GHz processor with Installed RAM: 8.00 GB. Each of the experiments are evaluated through various performance metrics including the Precision, Recall, F-measure and Accuracy.

Area under ROC curve is also used as a performance evaluation metric since it has the capability of predicting the probabilities of a tuple belonging to each target class instead of direct class prediction. Also the ROC curve is more suitable when the dataset has balanced classification. The points on the ROC are obtained as follows: ROC curve is plotted using the points obtained through false positive rate (x-axis) versus the true positive rate (y-axis) with respect to different candidate threshold values between 0.0 and 1.0.

IV. RESULTS AND DISCUSSION

The results of various experiments employing the different classifiers with proposed feature engineering pipelines(FEP) is described in multiple subsections. In section IV A (row wise analysis of Table 6), the performance evaluation of the each of the proposed FEPs across different classifiers is presented. The classifier resulting in highest accuracy with each pipeline is reported (highlighted in green color). Following this, section IV B (column wise analysis of Table 6) presents the performance evaluation of each of the classifier with different FEPs and reports the optimal FEP for each classifier that achieves highest accuracy with minimal features (highlighted in orange color). Section IV C and IV D include the performance metrics like Precision, Recall, F1 Score and Area under ROC. In section IV C, the metrics are reported for the stacking ensemble with different FEPs. Section IV D presents metrics for each classifier with the its respective optimal FEP. Section IV E presents the statistical significance tests followed by section IV F which presents the optimal

TABLE 6. Experimental results with different feature engineering pipelines.

Feature Engineering Approach	No of Features	Accuracy								
		NB	KNN	LR	SVM	DT	RF	AB	GB	Stack
None	31	89.10	89.29	92.69	61.54	96.15	97.94	92.63	97.17	98.12
CS	19	89.29	94.67	92.56	96.85	96.85	98.33	92.56	96.66	98.46
Filter	25	89.61	94.42	91.41	96.73	95.51	97.56	91.60	96.73	97.37
Filter(25)+CS	13	88.33	94.87	91.21	96.66	96.09	97.82	90.45	96.41	97.95
SVD (13)	13	87.82	92.69	92.69	98.07	96.09	97.24	95.83	96.47	98.27
SVD(20)+ CS	13	87.88	94.42	90.89	98.33	95.70	96.85	96.41	96.53	98.27
SVD(13)+CS	9	86.85	94.55	91.79	97.95	95.70	96.47	96.09	96.21	97.95
SVD(9)	9	88.65	91.98	89.80	95.12	94.35	95.96	94.67	95.57	96.98
SVD(13)+ Filter(8)	8	87.82	92.69	92.69	98.07	96.53	96.85	95.83	96.28	98.20
SVD(8)	8	86.74	90.96	89.80	94.29	94.67	96.34	95.19	95.25	96.60
SVD(13)+ Filter(10)+CS	6	88.56	93.80	90.19	94.76	95.39	96.42	96.10	96.17	97.18

Filter(x) : ANOVA Filter selecting x features, SVD(x) : Singular Value Decomposition with x features extracted

ML pipelines giving the best performance and section IV G presents the comparative analysis with the related work employing the same dataset. Finally, section IV H presents the limitations of the proposed method.

Table 6 shows the results of the experiments employing the different feature engineering pipelines with the different classifiers independently and finally a stacking ensemble utilizing these heterogeneous classifiers.

For each of the feature engineering pipeline, we have set a threshold for the highest classification accuracy obtained through either any of the eight classifiers or the stacking ensemble to be 96%. Hence, it can be observed that, for each row in table 6, the highest accuracy across all columns is always above 96%. The design of each of the feature engineering pipelines intends to achieve the accuracy above the threshold of 96% with minimized feature set size. Hence it can be observed that the successive feature engineering pipelines within each row strive to have a feature set size less than or equal to their predecessors while maintaining the highest accuracy threshold.

Also, the accuracy reported for each experiment is computed using the average obtained through 20 runs attributed to the stochastic nature of the algorithms.

A. PERFORMANCE ANALYSIS OF FEATURE ENGINEERING PIPELINES WITH DIFFERENT CLASSIFIERS

As mentioned in section III, the design criterion for the various feature engineering pipelines is to achieve an accuracy above 96% with minimal sized feature subset. As can be seen from table 6, the stacking ensemble classifier produces an accuracy of 98.12% with full feature set sized 31 features whereas the crow search based feature selection with a stacking ensemble produces an accuracy of 98.42% with reduced feature set sized 19 features. Apart from this, two other feature engineering pipelines produce an accuracy of 98.27% through stacking ensemble. The first one involves SVD with 13 components thereby resulting in a reduced feature set sized 13 features and the second one involves SVD with 20 components followed by Crow Search thereby generating a reduced feature subset of 13 features. Yet another feature engineering pipeline produces an accuracy of 98.2% with a stacking ensemble using a very small sized feature subset with just 8 features. It uses SVD with 13 components followed by ANOVA filter with 8 components.

Hence any of the above mentioned feature engineering pipelines can be employed with a stacking ensemble classifier (built using various heterogeneous base learners as shown in Fig. 2) to achieve an accuracy above 98% with reduced feature subset.

The classification accuracy is reported for each experiment along with the size of the feature set employed in Table 6. It can be observed that different models achieve highest accuracy with varying feature engineering pipelines. The stacking ensemble provides highest accuracy for all of the experiments except for the one where the ANOVA filter is applied for selecting 25 features. All the classifiers except the Gradient Boosting (GB) has improved accuracy with one of feature engineering pipelines. The GB classifier has the highest accuracy of 97.17% when all the 31 features are used for classification with no feature engineering applied.

From the second row, it can be observed that there is a significant improvement in the accuracy for all the classifiers when Crow Search (CS) is used for feature selection.

The CS selects an average of 19 features out of 31 features and all the classifiers show a boost in the accuracy with the reduced feature set. The highest boost in the accuracy is observed for SVM wherein a substantial increase of 35% is achieved when the CS algorithm is applied for feature selection.

When ANOVA filter is applied for feature selection with 25 features selected out of 31, the highest accuracy of 97.56% with the random forest classifier. In the third experiment, the feature engineering pipeline comprised the application of ANOVA filter for the selection of 25 features followed by CS algorithm for further feature reduction.

The average number of features chosen by the pipeline is 13 and the highest accuracy of 97.95% is achieved with the stacking ensemble.

The next experiment involves the usage of SVD technique for feature extraction out of the original feature set of 31 features. Compared to the earlier experiment, an extraction of 13 features through SVD results in the highest accuracy of 98.27% with the stacking ensemble.

Subsequent experiment has the same sized reduced feature set of 13 features but this time, the pipeline comprised feature extraction with SVD for 20 features followed by the CS algorithm for feature selection.

The highest accuracy of this experiment remains the same as the earlier one. Further reduction in the feature subset size is attempted through the next experiment ensuring that the highest accuracy does not fall below the threshold. The pipeline employs the SVD for feature extraction with 13 features followed by further reduction through CS algorithm. The feature subset size is reduced to 9 features on an average and the highest accuracy of 97.5% is achieved through stacking ensemble. We tried to analyze the performance with the same feature subset size of 9 features as the earlier experiment with a feature engineering pipeline involving the SVD in solo (without using the CS in succession).

It was found that the accuracy of all the classifiers was reduced (except the NB) compared to the previous experiment with the same feature subset size. For NB classifier, there was a considerable rise of around 2% in accuracy compared to the earlier experiment.

In the following experiment, feature subset size is reduced further while maintaining the highest accuracy threshold by using a feature engineering pipeline with SVD for feature extraction of 13 features followed by ANOVA filter for feature selection of 8 features. The reduced feature subset size is 8 and the highest accuracy of 98.2% is achieved. The performance with the same sized feature subset of 8 features obtained through SVD is found to be considerably reducing the accuracy across all the classifiers with an amount of around 2%.

The last experiment intends to further reduce the feature subset size from 8 features while maintaining the highest accuracy threshold. The feature engineering pipeline involved SVD for feature extraction with 13 features followed by ANOVA filter for feature selection with 10 features followed by CS algorithm. The resulting feature set size is 6 features which gives the highest accuracy of 97.18% with the stacking ensemble.

B. PERFORMANCE ANALYSIS OF CLASSIFIERS WITH DIFFERENT FEATURE ENGINEERING PIPELINES

For each classifier, we compare the highest accuracy and the reduced feature subset size obtained through one of the feature engineering pipelines with the corresponding values obtained when no feature engineering is employed (first experiment with full feature set sized 31 features).

For the NB classifier, it can be observed that the highest accuracy of 89.61% is achieved when the ANOVA filter is used for feature selection with 25 features. There is no significant impact upon accuracy upon the usage of any of the feature engineering pipelines. The KNN classifier achieves the highest accuracy of 94.87% with ANOVA filter for the selection of 25 features followed by the CS algorithm resulting in feature subset of only 13 features. It can be observed that there is a considerable improvement of around 6% in accuracy when compared to the classification employing no feature engineering. (first experiment with accuracy 89.29%).

For Logistic Regression, the highest accuracy of 92.69% is achieved in two experiments: firstly, when no feature engineering is applied with full feature subset of 31 features and secondly when feature engineering involving SVD with 13 features and ANOVA filter with 8 features is applied. Hence the latter approach is preferable as the highest accuracy is obtained with a considerably smaller feature subset involving only 8 features. The SVM classifier has the highest improvement on accuracy when the feature engineering is applied. It achieves the highest accuracy of 98.33% when a pipeline of SVD with 20 features followed by CS algorithm is used resulting in a feature subset of 13 features. This is a drastic increase of around 36.7% in accuracy when compared to the classification with no feature engineering.

In the Decision tree classifier, the highest accuracy of 96.85% is achieved with CS algorithm using a reduced feature subset of 19 features. The rise in accuracy is 0.7% which is not very significant compared to the accuracy with no feature engineering. The RF classifier achieves the highest accuracy with CS algorithm using reduced feature subset size of 19 features. Compared to the first experiment, the rise in accuracy is just 0.4% but the reduction in feature subset size is good from 31 features to 19 features. In the AB classifier, the highest accuracy of 96.41% is obtained with SVD generating 20 features followed by the CS algorithm resulting in a reduced feature subset of 13 features. This is a considerable rise in accuracy of around 4% compared to the approach with no feature engineering.

The GB classifier is the only one which shows no improvement in accuracy when any of the feature engineering pipelines are employed. It has the highest accuracy of 97.17% obtained with no feature engineering. Finally, the stacking ensemble classifier, obtains the highest accuracy of 98.46% with the CS algorithm resulting in reduced feature subset of 19 features.

C. EVALUATION METRICS FOR STACKING ENSEMBLE WITH DIFFERENT FEATURE ENGINEERING PIPELINES

The values for the different metrics of Precision, Recall, and F1 score is reported for the stacking ensemble using different FEPs in Table 7. For all the FEPs, the area under ROC is 0.99 ± 0.01.

For the Precision metric, the highest precision of for class 0 is achieved with CS algorithm with a value of 0.99. For class 1, the highest precision is 0.99 with SVD generating 20 features followed by CS algorithm resulting in 13 features.

For the Recall metric, the highest value for class 0 is 0.98 which is achieved with SVD generating 20 features followed by CS algorithm resulting in 13 features. For class 1, the

TABLE 7. Experimental results with different feature engineering pipelines through a stacking ensemble.

Feature Engg. approach	No. of features	Precision		Recall		F1 Score	
		0	1	0	1	0	1
None	31	0.98	0.98	0.97	0.99	0.98	0.99
CS	19	0.99	0.98	0.96	0.99	0.98	0.99
Filter	25	0.97	0.98	0.96	0.98	0.97	0.98
Filter(25)+CS	13	0.97	0.98	0.96	0.98	0.97	0.98
SVD (13)	13	0.98	0.98	0.97	0.99	0.98	0.99
SVD(20)+ CS	13	0.98	0.99	0.98	0.99	0.98	0.99
SVD(13)+CS	9	0.98	0.98	0.97	0.99	0.98	0.99
SVD(9)	9	0.97	0.97	0.95	0.98	0.96	0.98
SVD(13)+Filter(8)	8	0.98	0.98	0.97	0.99	0.98	0.99
SVD(8)	8	0.96	0.97	0.95	0.97	0.96	0.97
SVD(13)+Filter(10)+CS	6	0.97	0.98	0.97	0.98	0.97	0.98

TABLE 8. Performance of different classifiers with optimal feature engineering pipelines.

Classifier Name	FEP	FSS	Accuracy with FEP	Accuracy with no FEP	Area under ROC with FEP	Precision with FEP		Recall with FEP		F1-Sore with FEP	
						0	1	0	1	0	1
NB	Filter(25)	25	89.61	89.10	0.95±0.02	0.86	0.92	0.87	0.91	0.87	0.92
KNN	Filter(25)+CS	13	94.87	89.29	0.98±0.02	0.98	0.98	0.97	0.99	0.97	0.98
LR	SVD(13)+Filter(8)	8	92.69	92.69	0.97±0.02	0.91	0.94	0.90	0.94	0.90	0.94
SVM	SVD(20)+CS	13	98.33	61.54	0.99±0.01	0.99	0.98	0.96	0.99	0.98	0.99
DT	CS	19	96.85	96.15	0.96±0.03	0.96	0.97	0.96	0.97	0.96	0.97
RF	CS	19	98.33	97.94	0.99±0.00	0.98	0.99	0.98	0.98	0.98	0.98
AB	SVD(20)+CS	13	96.41	92.63	0.98±0.02	0.95	0.97	0.95	0.97	0.95	0.97
GB	None	31	97.17	97.17	0.99±0.01	0.96	0.98	0.97	0.97	0.96	0.98
Stack	CS	19	98.46	98.12	0.99±0.00	0.99	0.98	0.96	0.99	0.98	0.99

FEP: Feature Engineering Pipeline FSS: Feature Set Size

highest recall is 0.99 with SVD generating 13 features followed by ANOVA filter selecting 8 features resulting in reduced feature subset of 8 features. For the f1-score metric, the highest value for class 0 is 0.98 and the highest value for class 1 is 0.99 which is achieved with SVD generating 13 features followed by ANOVA filter selecting 8 features.

For class 1, the highest recall is 0.99 with SVD generating 13 features followed by ANOVA filter selecting 8 features resulting in reduced feature subset of 8 features.

For the F1-score metric, the highest value for class 0 is 0.98 and the highest value for class 1 is 0.99 which is achieved with svd generating 13 features followed by anova filter selecting 8 features.

D. EVALUATION METRICS FOR DIFFERENT CLASSIFIERS WITH OPTIMAL FEATURE ENGINEERING PIPELINES

Table 8 shows the optimal performance of the different classifiers with appropriate feature engineering pipelines resulting in a reduced feature set size. The table highlights the fact that the accuracy with each of the classifiers is improved when feature engineering pipeline (FEP) is used. For each performance metric, the highest value with minimal features is highlighted with green color. For area under ROC curve, the highest value of 0.99 is achieved with SVM using a FEP with SVD for feature extraction with 20 features followed by CS generating a total of 13 features. The same model generates the highest precision of 0.99 for class 0. For class 1, highest precision of 0.99 is obtained with RF classifier employing a FEP with CS generating 19 features. It also achieves the highest recall of 0.98 with class 0. For class 1, highest recall of 0.99 is obtained with two models: KNN classifier with FEP of ANOVA filter generating 25 features followed by CS resulting in a total of 13 features and SVM classifier with FEP of SVD for extracting 20 features followed by CS resulting in a total of 13 features. The highest value of F1-Score for class 0 and class 1 have the values of 0.98 and 0.99 respectively using the SVM classifier with FEP of SVD generating 20 features followed by CS resulting in 13 features.

Fig. 5 shows the ROC curves of each of the classifier with FEP giving the optimal performance. It can be observed that KNN with 13 features, SVM with 13 features, RF with 19 features, AB with 13 features, GB with 31 features and Stacking Ensemble with 19 features gives very good performance with area under ROC (AUC) greater than or equal to 0.98 with reduced feature set proving the efficiency of the proposed feature engineering pipelines.

E. STATISTICAL SIGNIFICANCE TESTS

The process of model selection involves the evaluation of different machine learning algorithms followed by their performance comparison. The model which gives the best value

TABLE 9. Experimental results of modified paired t-test.

Feature Engineering Approach	No of Features	p-value of each classifier in comparison with Stacking Ensemble							
		NB	KNN	LR	SVM	DT	RF	AB	GB
CS	19	0.027	0.031	0.022	0.031	0.011	0.032	0.013	0.015
SVD (13)	13	0.032	0.023	0.046	0.043	0.041	0.028	0.027	0.021
SVD(20)+ CS	13	0.043	0.025	0.027	0.047	0.036	0.047	0.045	0.027
SVD(13)+ Filter(8)	8	0.027	0.032	0.031	0.023	0.027	0.012	0.017	0.029

for the performance measure is chosen. But, the difference in the performance metric values is not real and sometimes attributed to statistical fluke [54]. Such a problem can be solved through a statistical hypothesis test.

As per Thomas Dietterich, the author of the widely cited research [55] on the comparison of classifiers through statistical significance tests, the McNemar's test [56] is specifically recommended when there is a separate unique test dataset being used for evaluation rather than resampling methods like K-Fold Cross Validation involving multiple evaluations.

Another research [57] explains that in the absence of an independent test data and the usage of resampling techniques like cross validation, the instability due to the training on 1-1/k subset of the dataset is directly related to the performance uncertainty of the model which can be estimated from repeated k-fold cross validation. This can be followed by a paired t-test.

When K-fold cross-validation is used, each ML model can be evaluated on the same split followed by the calculation of a score on each split. The comparison between the scores through a paired statistical hypothesis test like the paired student's t-test can be performed. But, the assumption of having independent model evaluations is not satisfied as the training of models is done using the same data subsamples/folds except when the records are chosen from the fold having hold-out data. Hence, the paired Student's t-Test is optimistically biased due to the lack of independence.

The modification of the paired t-test to take into consideration the lack of independence can be done using an approach proposed by Thomas Dietterich called as 5×2-fold cross-validation. The implementation of this method is provided through MLxtend library using the paired_ttest_5×2cv() function. The function accepts the data and models to generate the values of t-statistic and p-value. The p-value represents the statistical significance of the difference between the performance measures of the two models. Its current value is compared with a value termed as alpha which represents the threshold significance level (usually set to 0.05). Specifically, if the p-value is less or equal to the value of alpha, the null hypothesis meaning the same mean performance of the models is rejected which indicates that the difference in the performance is probably real. Otherwise, the null hypothesis cannot be rejected and any difference in the performance is not real and it could be attributed to statistical fluke.

Table 9 below shows the p-value obtained after applying modified t-test for each pair of classifiers involving the stacking ensemble and one of the remaining classifiers. Four feature engineering pipelines which give the highest accuracy above 98% have been chosen. It can be observed that all the p-values are less than the 0.05 threshold thereby resulting in the rejection of null hypothesis and proving the better performance of stacking ensemble in comparison with other classifiers.

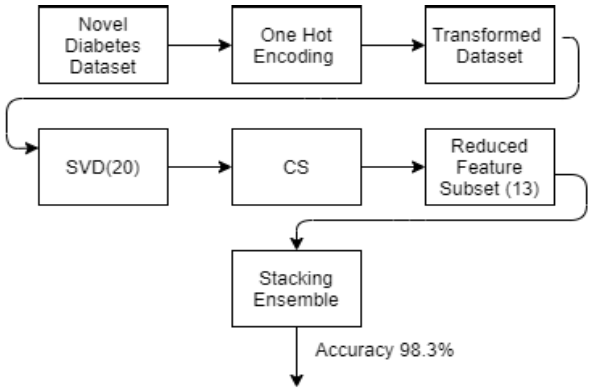

FIGURE 3. ML pipeline 1 with optimal performance.

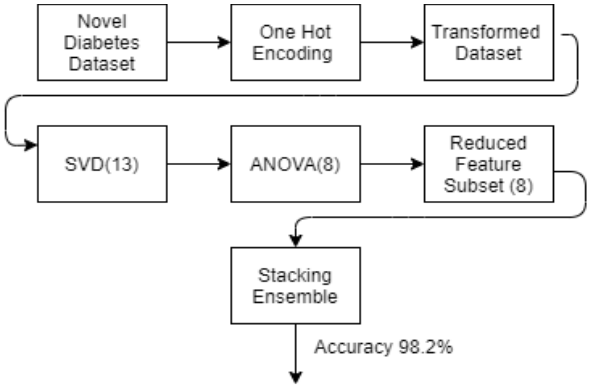

FIGURE 4. ML pipeline 2 with optimal performance.

F. MACHINE LEARNING PIPLEINES WITH OPTIMAL PERFORMANCE

After thorough experimental analysis, it was found that two ML pipelines provide the most optimal performance with the following characteristics:

ML Pipeline1: Feature subset size 13 and accuracy of 98.3% as shown in Fig. 3. The feature engineering with SVD extracts 20 most representative features out of which

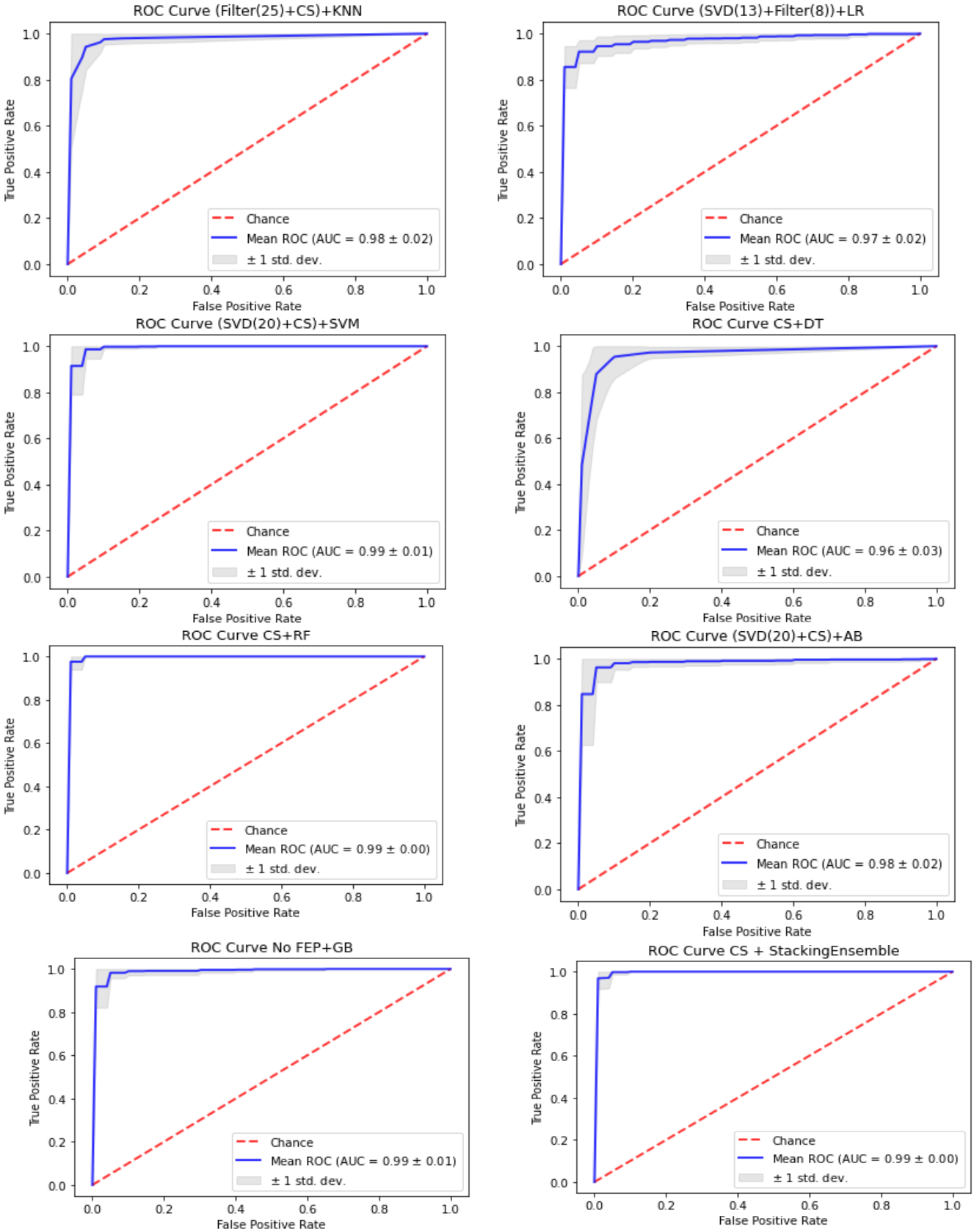

FIGURE 5. ROC curve for different classifiers with their respective optimal feature engineering pipeline.

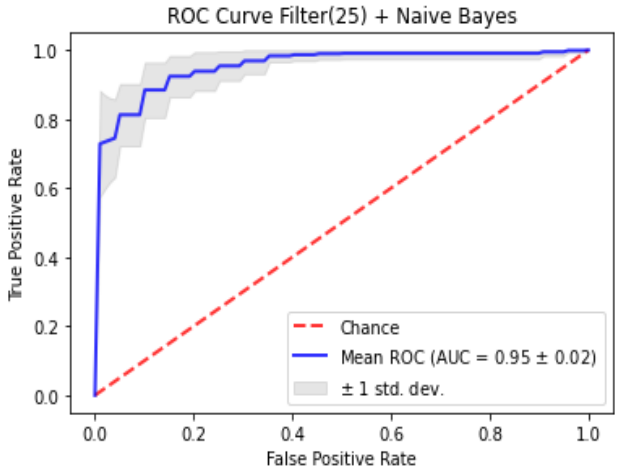

FIGURE 5. *(Continued.)* **ROC curve for different classifiers with their respective optimal feature engineering pipeline.**

TABLE 10. Comparative analysis of the experimental results with the related work using the same dataset.

Classifier Name	ML model in [8]			ML model our approach using FEP		
	ACC	FEP	FSS	ACC	FEP	FSS
NB	87.4	NA	31	89.61	Filter	25
LR	92.4	NA	31	92.69	SVD(13)+Filter(8)	8
DT	95.6	NA	31	96.85	CS	19
RF	97.4	NA	31	98.33	CS	19

FEP: Feature Engineering Pipeline FSS: Feature Set Size

feature selection is done to pick 13 most relevant features using the CS algorithm. Hence these two successive steps in the pipeline transform the dataset into its most representative form such that a stacking ensemble generates a classification accuracy of 98.3% using the reduced feature subset.

ML Pipleline2: Feature Subset size and accuracy of 98.2% as shown in Fig. 4. The feature engineering with SVD extracts 13 most representative features out of which feature selection is done to pick 8 most relevant features using the ANOVA filter.

Hence these two successive steps in the pipeline transform the dataset into its most representative for such that a stacking ensemble generates a classification accuracy of 98.2% using the reduced feature subset.

G. COMPARATIVE ANALYSIS WITH RELATED WORK

The performance of the proposed ML framework is compared with the ML model proposed in [8] as it employs the same diabetic dataset. Table 10 shows the result. For all the classifiers employed in [8], our approach employing various feature engineering pipelines gives better accuracy with reduced feature set size.

H. LIMITATIONS OF THE PROPOSED METHOD

Some of the limitations of the proposed ML pipelines are as follows:

i) When CS is employed for feature selection, the execution time might be relatively higher owing to the higher number of iterations in the CS algorithm for a better classification accuracy.

ii) Even though the usage of SVD for feature extraction provides a significant performance boost with the current dataset, one trade-off worth noting is the loss of feature interpretability. This might be a deterrent in the medical domain where the influence of a feature upon the machine learning model drives the following steps.

V. CONCLUSION AND FUTURE WORK

In the current research, a novel diabetes dataset from the UCI repository is employed rather than the benchmark PID dataset.

As the PID dataset is restricted to female patients, the intent of the usage new dataset was to be applicable to any gender and work towards early diagnosis by collecting simple binary values for different attributes representing common and rare symptoms related to the onset of diabetes.

We design various feature engineering pipelines to reduce the feature set size while maintaining high accuracy. The stacking ensemble has the ability to harness the skills of multiple heterogeneous models and generate an improved performance. Hence the proposed work employs the stacking ensemble for the evaluation of various feature engineering pipelines.

The highest accuracy of 98.46% was obtained with CS algorithm for feature selection resulting in 19 features using a stacking ensemble model. Almost same accuracy is also achieved with still smaller feature set size of 13 and 8 features with feature engineering pipelines of SVD (20) +CS and SVD (13) +Filter (8) respectively.

The proposed ML pipeline is effective on all data samples which can be justified as follows: The performance evaluation metrics reported for each experiment are computed using the average obtained through 20 runs attributed to the stochastic nature of the algorithms. The train test split followed by repeated stratified K-fold used for generating train, validation and test data sets ensures that each run has a unique data samples.

The future work involves the application of the proposed predictive models with the feature engineering pipelines to other higher dimensional datasets which may belong to other/medical context for accurate diagnosis with reduced storage and computational costs.

ACKNOWLEDGMENT

Shirina Samreen would like to thank Deanship of Scientific Research at Majmaah University for supporting this work under Project Number R-2021-224.

REFERENCES

[1] The 6 Different Types of Diabetes: (Mar. 5, 2018). *The Diabetic Journey*. [Online]. Available: https://thediabeticjourney.com/the-6-different-types-of-diabetes

[2] (Jul. 8, 2018). *Failure to Detect Type 2 Diabetes Early Costing 700 Million Per Year, Diabetes Australia*. [Online]. Available: https://www.diabetesaustralia.com.au

[3] M. I. Harris, R. Klein, T. A. Welborn, and M. W. Knuiman, "Onset of NIDDM occurs at least 4-7 yr before clinical diagnosis," *Diabetes Care*, vol. 15, no. 7, pp. 815–819, Jul. 1992.
[4] N. Cho, J. E. Shaw, S. Karuranga, Y. D. Huang, J. D. da Rocha Fernandes, A. W. Ohlrogge, and B. Malanda, "IDF diabetes atlas: Global estimates of diabetes prevalence for 2017 and projections for 2045," *Diabetes Res. Clin. Pract.*, vol. 138, pp. 271–281, Apr. 2018.
[5] N. Sarwar, P. Gao, S. R. K. Seshasai, R. Gobin, S. Kaptoge, E. D. Angelantonio, E. Ingelsson, D. A. Lawlor, E. Selvin, M. Stampfer, C. D. A. Stehouwer, S. Lewington, L. Pennells, A. Thompson, N. Sattar, I. R. White, K. K. Ray, and J. Danesh, "Diabetes mellitus, fasting blood glucose concentration, and risk of vascular disease: A collaborative meta- analysis of 102 prospective studies," *Lancet*, vol. 375, no. 9733, pp. 2215–2222, Jul. 2010.
[6] (Mar. 22, 2018). *Statistics About Diabetes: American Diabetes Association*. [Online]. Available: https://www.diabetes.org
[7] S. Akter, M. M. Rahman, S. K. Abe, and P. Sultana, "Prevalence of diabetes and prediabetes and their risk factors among Bangladeshi adults: A nationwide survey," *Bull. World Health Organ*, vol. 92, pp. 213–320, Mar. 2014.
[8] M. F. Islam, R. Ferdousi, S. Rahman, and H. Y. Bushra, "Likelihood prediction of diabetes at early stage using data mining techniques," in *Computer Vision and Machine Intelligence in Medical Image Analysis*. Singapore: Springer, 2020, pp. 113–125.
[9] G. I. Webb, J. R. Boughton, and Z. Wang, "Not so naive Bayes: Aggregating one-dependence estimators," *Mach. Learn.*, vol. 58, no. 1, pp. 5–24, Jan. 2005.
[10] P. Cunningham and S. J. Delany, "k-nearest neighbour classifiers," *Multiple Classifier Syst.*, vol. 34, pp. 1–17, Mar. 2007.
[11] B. P. Tabaei and W. H. Herman, "A multivariate logistic regression equation to screen for diabetes : Development and validation," *Diabetes Care*, vol. 25, no. 11, pp. 1999–2003, Nov. 2002.
[12] I. Jenhani, N. B. Amor, and Z. Elouedi, "Decision trees as possibilistic classifiers," *Int. J. Approx. Reasoning*, vol. 48, no. 3, pp. 784–807, Aug. 2008.
[13] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, pp. 237–297, Sep. 1995.
[14] B. Kégl, "The return of AdaBoost.MH: Multi-class Hamming trees," 2013, *arXiv:1312.6086*. [Online]. Available: http://arxiv.org/abs/1312.6086
[15] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[16] L. Li, "Diagnosis of diabetes using a weight-adjusted voting approach," in *Proc. IEEE Int. Conf. Bioinf. Bioeng.*, Nov. 2014, pp. 320–324.
[17] A. Reinhardt, "Using neural networks for prediction of the subcellular location of proteins," *Nucleic Acids Res.*, vol. 26, no. 9, pp. 2230–2236, May 1998.
[18] M. Pradhan and G. R. Bamnote, "Design of classifier for detection of diabetes mellitus using genetic programming," in *Proc. 3rd Int. Conf. Frontiers Intell. Comput., Theory Appl.*, Nov. 2015, pp. 763–770.
[19] N. Nai-arun and R. Moungmai, "Comparison of classifiers for the risk of diabetes prediction," *Proc. Comput. Sci.*, vol. 69, pp. 132–142, Nov. 2015.
[20] S. Perveen, M. Shahbaz, A. Guergachi, and K. Keshavjee, "Performance analysis of data mining classification techniques to predict diabetes," *Proc. Comput. Sci.*, vol. 82, pp. 115–121, Mar. 2016.
[21] S. Brahim-Belhouari and A. Bermak, "Gaussian process for nonstationary time series prediction," *Comput. Statist. Data Anal.*, vol. 47, no. 4, pp. 705–712, 2004.
[22] M. Maniruzzaman, N. Kumar, M. M. Abedin, M. S. Islam, H. S. Suri, A. S. El-Baz, and J. S. Suri, "Comparative approaches for classification of diabetes mellitus data: Machine learning paradigm," *Comput. Methods Programs Biomed.*, vol. 152, pp. 23–34, Dec. 2017.
[23] T. Sapatinas, "Discriminant analysis and statistical pattern recognition," *J. Roy. Stat. Soc. A, Statist. Soc.*, vol. 168, no. 3, pp. 635–636, 2005.
[24] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. EC-14, no. 3, pp. 326–334, Jun. 1965.
[25] D. Sisodia and D. S. Sisodia, "Prediction of diabetes using classification algorithms," *Proc. Comput. Sci.*, vol. 132, pp. 1578–1585, Jan. 2018.
[26] Q. Wang, W. Cao, J. Guo, J. Ren, Y. Cheng, and D. N. Davis, "DMP_MI: An effective diabetes mellitus classification algorithm on imbalanced data with missing values," *IEEE Access*, vol. 7, pp. 102232–102238, 2019.
[27] A. K. Dewangan and P. Agrawal, "Classification of diabetes mellitus using machine learning techniques," *Int. J. Engg. Appl. Sci.*, vol. 2, no. 5, pp. 145–148, May 2015.
[28] S. Bashir, U. Qamar, and F. H. Khan, "IntelliHealth: A medical decision support application using a novel weighted multi-layer classifier ensemble framework," *J. Biomed. Inform.*, vol. 59, pp. 185–200, Feb. 2016.
[29] R. Vaishali, R. Sasikala, S. Ramasubbareddy, S. Remya, and S. Nalluri, "Genetic algorithm based feature selection and MOE fuzzy classification algorithm on Pima Indians diabetes dataset," in *Proc. Int. Conf. Comput. Netw. Informat. (ICCNI)*, Oct. 2017, pp. 1–5, doi: 10.1109/ICCNI.2017.8123815.
[30] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA, USA: Addison-Wesley, 1989.
[31] M. Maniruzzaman, M. J. Rahman, M. Al-MehediHasan, H. S. Suri, M. M. Abedin, A. El-Baz, and J. S. Suri, "Accurate diabetes risk stratification using machine learning: Role of missing value and outliers," *J. Med. Syst.*, vol. 42, no. 5, pp. 1–17, May 2018, doi: 10.1007/s10916-018-0940-7.
[32] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[33] D. Howell, *Statistical Methods for Psychology*. Davis Drive Belmont, CA, USA: Wadsworth Publishing, Jun. 2001.
[34] C. R. Rao, "The use and interpretation of principal component analysis in applied research," *Sankhyā, Indian J. Statist., Ser. A*, vol. 26, pp. 329–358, Dec. 1964.
[35] H. Kaur and V. Kumari, "Predictive modelling and analytics for diabetes using a machine learning approach," *Inf. Technol. Ind.*, vol. 9, no. 1, pp. 215–223, Feb. 2021.
[36] T. Li and S. Fong, "A fast feature selection method based on coefficient of variation for diabetics prediction using machine learning," *Int. J. Extreme Autom. Connectivity Healthcare*, vol. 1, no. 1, pp. 55–65, Jan. 2019.
[37] N. Sneha and T. Gangil, "Analysis of diabetes mellitus for early prediction using optimal features selection," *J. Big Data*, vol. 6, no. 1, pp. 1–9, Dec. 2019, doi: 10.1186/s40537-019-0175-6.
[38] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, pp. 76516–76531, 2020, doi: 10.1109/ACCESS.2020.2989857.
[39] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Netw.*, vol. 13, nos. 4–5, pp. 411–430, Jun. 2000.
[40] F. Han and H. Liu, "Statistical analysis of latent generalized correlation matrix estimation in transelliptical distribution," *Bernoulli*, vol. 23, no. 1, pp. 23–57, Feb. 2017.
[41] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.
[42] B. Zolghadr-Asli, O. Bozorg-Haddad, and X. Chu, "Crow search algorithm (CSA)," in *Advanced Optimization by Nature-Inspired Algorithms, Studies in Computational Intelligence*, vol. 720. Singapore: Springer, 2018, pp. 143–149.
[43] V. Klema and A. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Trans. Autom. Control*, vol. AC-25, no. 2, pp. 164–176, Apr. 1980.
[44] L.-Y. Chuang, C.-H. Yang, K.-C. Wu, and C.-H. Yang, "A hybrid feature selection method for DNA microarray data," *Comput. Biol. Med.*, vol. 41, no. 4, pp. 228–237, Apr. 2011.
[45] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Appl. Soft Comput.*, vol. 18, pp. 261–276, May 2014.
[46] E. Hancer, B. Xue, D. Karaboga, and M. Zhang, "A binary ABC algorithm based on advanced similarity scheme for feature selection," *Appl. Soft Comput.*, vol. 36, pp. 334–348, Nov. 2015.
[47] S. Chattopadhyay, S. Mishra, and S. Goswami, "Feature selection using differential evolution with binary mutation scheme," in *Proc. Int. Conf. Microelectron., Comput. Commun. (MicroCom)*, Jan. 2016, pp. 1–6.
[48] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa, and X.-S. Yang, "BBA: A binary bat algorithm for feature selection," in *Proc. 25th SIBGRAPI Conf. Graph., Patterns Images*, Aug. 2012, pp. 291–297.
[49] H. M. Zawbaa, E. Emary, B. Parv, and M. Sharawi, "Feature selection approach based on moth-flame optimization algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 4612–4617.
[50] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016.
[51] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Expert Syst. Appl.*, vol. 116, no. 1, pp. 147–160, 2019.

[52] M. A. Elaziz, D. Oliva, and S. Xiong, "An improved opposition-based sine cosine algorithm for global optimization," *Expert Sys. Appl.*, vol. 90, pp. 484–500, Dec. 2017.
[53] S. Ouadfel and M. A. Elaziz, "Enhanced crow search algorithm for feature selection," *Expert Syst. Appl.*, vol. 159, Nov. 2020, Art. no. 113572.
[54] *How to Calculate McNemar's Test to Compare Two Machine Learning Classifiers*. Accessed: Jul. 25, 2018. [Online]. Available: https://machinelearningmastery.com/mcnemars-test-for-machine-learning/
[55] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, 1998.
[56] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.
[57] C. Beleites and R. Salzer, "Assessing and improving the stability of chemometric models in small sample size situations," *Anal. Bioanal. Chem.*, vol. 390, no. 5, pp. 1261–1271, Mar. 2008.
[58] X. Li, S. Wang, and Y. Cai, "Tutorial: Complexity analysis of singular value decomposition and its variants," 2019, *arXiv:1906.12085*. [Online]. Available: http://arxiv.org/abs/1906.12085
[59] (Dec. 2019). *Computational Complexity of ML Models*. [Online]. Available: https://medium.com/analytics-vidhya/time-complexity-of-ml-models-4ec39fad2770
[60] (Apr. 2018). *Computational Complexity of Machine Learning Algorithms*. ttps://www.thekerneltrip.com/machine/learning/computational-complexity-learning-algorithms/

SHIRINA SAMREEN (Member, IEEE) received the Ph.D. degree in computer science and engineering from Jawaharlal Nehru Technological University Hyderabad (JNTUH), India, in 2016. She is currently an Assistant Professor with the Department of Computer Science, College of Computer and Information Sciences, Majmaah University, Saudi Arabia. She has 30 research papers to her credit in various IEEE international conferences/journals. She received best paper awards twice for her research papers at IEEE ICCIC, a renowned international conference. She is a reviewer of *IEEE Wireless Communications Magazine*, *Journal of Engineering and Applied Sciences* (JEAS), IEEE Access, and various IEEE International Conferences held in India and abroad.