IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 39, No. 1, February, 2022, pp. 87-107
Journal homepage: http://iieta.org/journals/ts

Improving Depression Prediction Accuracy Using Fisher Score-Based Feature Selection and Dynamic Ensemble Selection Approach Based on Acoustic Features of Speech

Naulegari Janardhan*, Nandhini Kumaresh

Department of Computer Science, School of Mathematics and Computer Sciences, Central University of Tamil Nadu, Thiruvarur 610101, India

Corresponding Author Email: janardhan17@students.cutn.ac.in

https://doi.org/10.18280/ts.390109

Received: 23 December 2021
Accepted: 13 February 2022

Keywords:
acoustic features, depression, dynamic ensemble selection, feature selection, fisher score, METADES, openSMILE, KNORAU

ABSTRACT

Depression affects over 322 million people, and it is the most common source of disability worldwide. Literature in speech processing revealed that speech could be used for detecting depression. Depressed individuals exhibit varied acoustic characteristics compared to non-depressed. A four-staged machine learning classification system is developed to investigate the acoustic parameters to detect depression. Stage one uses speech recordings from a publicly available and clinically validated dataset DAIC-WOZ. The baseline acoustic feature vector, eGeMAPS, is extracted from the dataset in stage two. Adaptive synthetic (ADASYN) is performed along with data preprocessing to overcome the class imbalance. In stage three, we conducted feature selection (FS) using three techniques; Boruta FS, recursive feature elimination using support vector machine (SVM-RFE), and the fisher score-based FS. Experimentation with various machine learning base classifiers like gaussian naïve bayes (GNB), support vector machine (SVM), *k*-nearest neighbors (KNN), logistic regression (LR), and random forest classifier (RF) is performed in stage four. The hyperparameters of the classifiers are tuned using the GridSearchCV technique throughout the 10-fold stratified cross-validation (CV). Then we employed multiple dynamic ensemble selection of classifier algorithms (DES) with k=3 and k=5 utilizing the pool of aforementioned four base classifiers to improve the accuracy. We present a comparative study using eGeMAPS features against the base classifiers and the experimented DES classifiers. Our results on the DAIC-WOZ benchmark dataset suggested that K-Nearest Oracles Union (KNORA-U) DES with k=3 has superior accuracy using a subset of 15 features selected by fisher score-based FS than the individual base classifiers.

1. INTRODUCTION

The prevalence of depressive disorders has increased since the last decade [1]. The World Health Organization (WHO) claims that depression is a widespread mental illness, and approximately 322 million people are suffering from it. WHO reports that depression is a primary cause of disability and suicides (about Eight lakh cases each year) [2]. Clinical depression is a pathology characterized by a decrease in positive emotions and feelings and a rise in negative emotions and feelings. Consequently, patients suffer from depressed mood, reduced motivation, lack of interest, shame or low self-esteem, poor concentration, and disruptive sleep or appetite [3]. It is suggested that around two-thirds of cases are going unidentified or undiagnosed. To help psychiatrists identify and diagnose depression effectively, there should be a methodology that takes advantage of artificial intelligence and advancements in machine learning.

Advances in artificial intelligence (AI) and machine learning (ML) have had a major impact in the medical field. To assist Psychiatrists and automate and speed up the entire diagnosis process, there has been much research into automated depression prediction in recent years. For example, neuroscientists can predict an 81 percent positive probability for autism using magnetic resonance imaging (MRI) and deep learning algorithms [4]. Psychologists are also using MRI, biomarkers, and audiovisual approaches to detect mental diseases such as depression using AI [5].

According to research by Sobin and Sackeim [6], depression manifests itself in behavioral changes in several daily activities and the way people communicate. The speech of a depressed person is consistently described by clinicians as repetitive, dull, and spiritless [7]. As a result, detecting depression based on acoustic aspects of a person's speech is a study topic for further investigation utilizing various machine and deep learning techniques. Several methods for determining the relationship between depression and acoustic variables are proposed to classify depressed voices [8].

Various works have explored individual classifiers and multiple classifiers selection for identifying depressed and non-depressed voices. However, there is still scope for improving the classification accuracy using dynamic ensemble classification. Previous studies have extracted various acoustic features of a person's speech; nevertheless, it is still inexact which acoustic characteristics are best suited for depression detection [9]. Moreover, despite ongoing research, accurately diagnosing depression with a minimal feature set using advanced ML techniques remains an unexplored task.

Therefore, this research focuses on using the selected baseline eGeMAPS acoustic features and advanced dynamic ensemble classifiers to improve the prediction performance of depressed and non-depressed. To achieve this, a four-staged

machine learning classification model is developed using the dynamic ensemble selection of classifiers, like Meta-Learning for DES (METADES), K-Nearest Oracles Eliminate (KNORA-E), KNORA-U, and DES for Multiclass Imbalance (DES-MI). The proposed work is validated on the DAIC-WOZ dataset. Features with 80 percent correlation among themselves and the constant features are deleted from the feature set before proceeding with the FS process. Such deletion will lead to eliminate the redundancy in the feature vector. The performance of the model has been improved after eliminating the correlated features. The accuracy and balanced accuracy was 61% and 67% using all the features, and 68% and 72% after eliminating the correlated features. After performing the FS, the performance is significantly improved. The accuracy of 82% and balanced accuracy of 77% is obtained using Fisher score-based FS.

The remaining paper is organized as: the immediate section focuses on the review of literature emphasizing the severity of depression and the possible speech biomarkers that can be used to detect it. It also describes the works done in FS and the classifiers transition to dynamic ensemble classifiers. Section 3 introduces our methodology, where the dataset, feature extraction, data preprocessing, FS, and classification are discussed. Section 4 presents our results and a brief discussion about the results. Section 5 discusses the conclusion and future research directions.

2. LITERATURE SURVEY

One of the emerging disciplines in human-computer interaction is Speech Emotion Recognition (SER). The types of characteristics used, as well as the classifiers used for recognition, have a significant impact on the quality of the human-computer interaction that imitates human speech emotions [10]. Aloshban et al. [11] revealed that combining linguistic and acoustic features of speech might be utilised to distinguish between depressed and non-depressed speakers with an accuracy of more than 80%. Lee et al. [12] found that males' spectral and energy-related acoustic features, and females' prosody-related acoustic features, were found to be strong discriminators for primary depressive disease and could be used as biomarkers for depression in the elderly. Samareh et al. [13] examined how depression will leave identifiable biomarkers in patients' acoustic, linguistic, and facial patterns. The authors broke the audio signal into 20-40 ms frames and extracted 35 audio biomarkers to capture the speaker's prosodic qualities and voice quality in the time and frequency domain. Low et al. [14] investigated five acoustic feature categories like Teager energy operator (TEO), glottal, spectral, and prosodic to detect clinical depression in adolescents. They collected 139 samples (68 clinically depressed and 71controls) in naturalistic interactions between parents and their adolescent children.

Toto et al. [15] proved that machine learning has the potential to aid psychotherapy by improving the effectiveness of mental health screening. They presented Sliding Window Sub-clip Pooling and an audio classification method for shorter datasets to tackle the depression screening from voice. Their work is also tested on the DAIC-WOZ database. Vázquez-Romero and Gallardo-Antolín [16] have proposed an automatic approach for determining whether a person is depressed by examining his or her voice in this research. It works on ensemble averaging, which combines M=50 One-Dimensional Convolutional Neural Networks in a single network (1d-CNN). Their system was tested on the DAIC-WOZ dataset as part of the Depression Sub-Challenge of The Audio/Visual Emotional Challenge workshop (AVEC-2016). It was compared to the baseline system based on a classifier like SVM and hand-crafted features. It is also based on the DepAudionet architecture, which includes 1d-CNN, Long Short-Term Memory (LSTM) recurrent neural network, and fully connected layers. According to the results, their system showed improved performance than the baseline, the DepAudionet, as well as the single 1d-CNN architecture by 58.5 percent, 30.0 percent, and 10.2 percent, respectively, in terms of F1-score.

FS plays a critical role in selecting effective features for classification, resulting in accuracy gain in any problem. FS also helps in the selection of effective biomarkers to detect depression. Usually, FS is carried out for these reasons: eliminate ambiguous data, reduce model training time, and avoid overfitting [17].

Many pieces of research are going on to find out the well-performing FS methods. Rong et al. [18] proposed Ensemble Random Forest to Tress (ERFTrees) to extract compelling features from small datasets. They proved by experiment that the dataset with a subset of 16 selected features could improve the accuracy compared to the base 84 feature set. Haider et al. [19] developed and compared their own FS method, Active Feature Selection (AFS), against three distinct state-of-the-art FS techniques: generalized Fisher score, ReliefF, and Infinite Latent Feature Selection (ILFS). They used eGeMAPS and emobase standard acoustic paralinguistic feature sets to evaluate EmoDB, SAVEE, and EMOVO emotion identification datasets. The findings revealed that by employing subsets of much smaller features than the complete feature set, either the same or improved accuracy could be attained.

Drotár et al. [20] presented multiple ensemble FS algorithms based on the schemes of voting aggregation like Borda count, new weighted Borda count, single transferable vote, and plurality vote. They also introduced a novel notion of clustering FS methods, finding that ensembles and clustered ensembles using a weighted Borda count perform exceptionally well.

Ensemble classification approaches have been widely studied in the fields of machine learning and artificial intelligence in recent years, both in industry and in the literature. Jiang et al. [9] conducted research involving 170 China-based volunteers (85 depressed participants and 85 healthy controls), where automatic depressed speech classification was studied. To detect depression, the classification performances of glottal, spectral, and prosodic speech variables have been examined. They also proposed an ensemble logistic regression model for depression detection (ELRDD) with LR as the base classifier. It produced encouraging results, with an improved accuracy rate of 75.00 percent for females and 81.82 percent for males, and a favorable sensitivity/specificity ratio of 79.25 percent/70.59 percent for females and 78.13 percent/85.29 percent for males.

Ostvar et al. [21] have presented a heterogeneous dynamic ensemble classifier (HDEC) that employs multiple classification algorithms trained with the training dataset. Later they separated the classifiers that are accurate in identifying the positive samples and the classifiers that are accurate in identifying the negative samples. To evaluate the HDEC, they have applied it on twelve standard datasets from

the repository of the University of California Irvine (UCI) and compared it against three state-of-the-art approaches in the ensemble technique, like Bagging, Boosting, and Stack Generalization, and achieved increased accuracy and geometric mean values.

After the extensive literature survey, it is found that the dynamic ensemble selection of classifier algorithms is not explored for depression detection exhaustively. Therefore this work focuses on using a dynamic ensemble of classifiers for depression detection with state-of-the-art techniques such as METADES, KNORAE, KNORAU, and DESMI with the pool of five base classifiers. The objective of this research is to improve the performance of depression detection using advanced machine learning techniques and several effective FS techniques using eGeMAPS features.

3. METHODOLOGY

The flow of the proposed methodology is depicted in Figure 1. It is divided into four stages. In stage one, the data is collected from the clinically validated DAIC-WOZ database. In stage two, the required features are extracted from the database, and preprocessing is done to make the extracted features' data more meaningful and relevant for the classification. Stage three contains the FS with different methods that are used to identify the suboptimal feature set. Finally, in stage four, the classification of depressed or non-depressed is achieved using the five base classifiers and the dynamic ensemble classifiers META-DES, KNORA-E, KNORA-U, and DES-MI. The subsequent sections elaborate on these four stages.

3.1 Data collection

This study employs The Distress Analysis Interview Corpus Wizard-of-Oz (DAIC-WOZ) for evaluating the performance of the proposed approach. The reason behind choosing this corpus in our work is the extensive usage of it as the benchmarking dataset by the research community in depression diagnosis. Several recent researchers have used this corpus for their works [15, 22-25]. DAIC-WOZ is a subset of the DAIC multimodal depression corpus. DAIC-WOZ is a publicly available clinically evaluated dataset recorded from the interview of the participants conducted by a virtual human agent, Ellie, which is controlled by a human interviewer in a different location. The interview lasted for 7 to 33 minutes (with an average of 16 minutes) for each participant. The questions and answers between the participant and Ellie were recorded with a high-performing and close-talk (fixed single-channel) microphone with negligible environmental background noise. It contains the data of 189 participants in total, that is partitioned into training (107 participants), development (35 participants), and testing (47 participants) partitions for AVEC 2016-17 challenges, as shown in Table 1.

Audio, video, and responses to the questionnaires were gathered, as well as transcripts of the interviews are made. The decision of whether the participant is depressed or non-depressed was taken based on the score of the individual participant on the Patient Health Questionnaire of eight items (PHQ-8) scale of depression [26]. In large-scale clinical investigations, PHQ-8 has been proven to be a valid diagnostic and severity measure for depressive disorders. The participant is considered depressed if the PHQ-8 score is >=10 and non-depressed if the PHQ-8 score is <10.

Table 1. Summary of the dataset

	Train Set	Development Set	Test Set	Sum
Non-depressed	77	23	33	133
Depressed	30	12	14	56
Total participants	107	35	47	189

3.2 Feature extraction

This work uses the openSMILE (open-source Speech and Music Interpretation by Large-space Extraction) toolkit to extract the features from the speech recordings of the DAIC-WOZ database. It is a tool for standardized audio feature extraction and classification [27]. To get the finite set of data to train the machine learning models, emotion identification uses smaller sets of knowledge-driven features like eGeMAPS [28]. Despite the lack of a universally accepted standard feature set, eGeMAPS has been used as a baseline feature set in AVEC since 2016 and is increasingly being used in recent research as the research presented at the Interspeech conference.

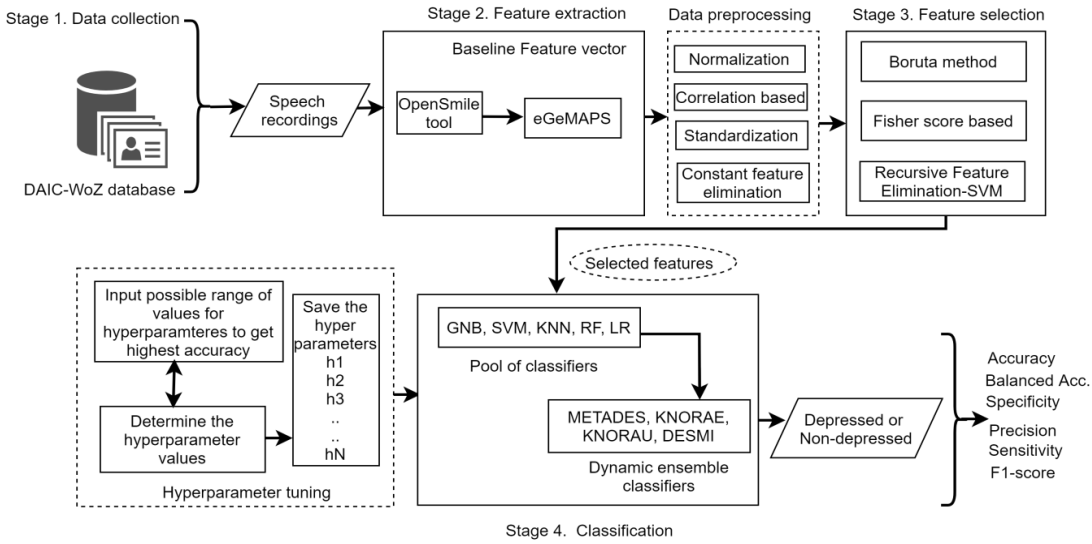

Figure 1. Overview of the proposed model

Table 2. Meaning of eGeMAPS features

Feature	Meaning
Fundamental frequency (F_0)	Frequency with which vocal folds snap; is considered as pitch
Jitter	F_0 variation from one period of speech to the next period
Shimmer	How much amplitude varies from one period of speech to the next
Loudness	Approximation of perceived signal intensity calculated as sum of spectrum mimicking human auditory perception
HNR- Harmonics to Noise Ratio	Degree of Periodicity of speech: $HNR = 10 \times log_{10} \frac{percentage\ of\ periodic\ wave}{percentage\ of\ noise}$; where a period is estimated from each 10 ms
Spectral slope (0-500 Hz and 500-1500 Hz)	The rate at which the amplitudes of consecutive frequencies in the spectrum decline as they become higher in frequency; this is referred to as voice timbre.
Alpha Ratio	Ratio between the sum of energy in low frequency region (50 to 1000 Hz) and in high frequency region (1 to 5 kHz)
Hammarberg Index	The ratio of the energy maxima in the 0 to 2 kHz and 2 to 5 kHz bands.
Formants 1–3 (F_1 to F_3) frequency	The frequencies where F1 to F3 are found (Formants are resonating frequencies of the vocal tract)
Formant 1 bandwidth	Frequency region around the F_1 frequency which is amplified
Formants 1–3 relative energy	Closest harmonic ratio of F_0 to the frequency of the formant. For Formant 1, it will be the ratio of the first formant
Harmonic difference (H1–H2)	Energy ratio of the first harmonic of F_0 to the second one (Harmonics are frequency bands at the multiples of F_0)
Harmonic difference (H1–A3)	Energy ratio of the first harmonic of F_0 to the highest harmonic in F_3 range

The (extended) Geneva Minimalistic Acoustic Parameter Set ((e)GeMAPS) feature set [28] was created to standardize affective computing research by generating the best collection of engineered features. The researchers chose the most effective features to design it based on three factors: a) whether a feature can show changes in voice output, b) how valuable a feature was in prior research, and c) its theoretical significance. Eyben et al. [28] proposed two versions of the eGeMAPS feature set; they are minimalistic and extended. The minimalistic feature set contains 18 spectral, voice quality, and amplitude low-level descriptors (LLDs) that are more effective by previous research. These LLDs are extracted at every 10 ms of the speech. eGeMAPS features and their brief description is provided in Table 2.

To obtain the utterance level functionals, the standard deviation and mean of the 18 LLDs are calculated, generating 36 features. Later, from loudness and fundamental frequency, the following features are calculated, such as 20^{th}, 50^{th}, and 80^{th} percentiles, range of 20^{th} to 80^{th} percentiles, standard deviation, and mean of the slope of rising and falling portion of the signal. As a result, 52 functionals are generated. The means of Alpha ratio, Hammarberg Index, and spectral slopes are also included. Therefore, in total, it contains 56 parameters. The minimalistic feature set consists of the rate of loudness peaks, i.e., the count of loudness peaks in a second, standard deviation, and mean length of continuous voiced speech. Continuous voiced speech is the speech delivered when the vocal folds vibrate. It contains the standard deviation and mean length of continuous unvoiced speech, i.e., the speech delivered when the vocal folds do not vibrate, and the number of continuous voiced regions per second. Therefore it has 62 parameters that constitute the GeMAPS minimalistic set.

The standard deviation and mean of the spectral flux and MFCCs (Mel Frequency Cepstral Coefficients) 1-4 in voiced regions only, as well as the standard deviation and mean of the spectral flux and MFCCs 1-4 in unvoiced regions only, are also presented. As a result, we get an extra 25 functionals. The equivalent sound level is also included. It's a feature that calculates the average quantity of background noise recorded. As a result, 88 functionals make up the eGeMAPS.

Machine learning models suffer from the dimensionality curse, which means that the accuracy of the predictions decreases as the number of features increases [29]. Usually, the majority of the features are unrelated to the classification, and hence their relevance is unknown in advance. Therefore selecting a small set of best features showing the best possible classification performance is advisable for practical reasons. This is achieved by several FS methods.

3.2.1 Data preprocessing

Normalization is performed to transform all the numeric columns to a common scale. In this work, as all the columns are numeric, the values are scaled-down between 0 and 1. The formula for normalization is given as Eq. (1):

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

If any two independent variables are correlated above the threshold, then they are considered duplicate variables, and one of the two variables should be eliminated. Here the features that are strongly correlated among themselves with more than 0.8 of correlation are eliminated using the Pearson correlation technique. The correlation coefficient threshold is considered as 0.8 depending upon the research works carried out in medical research [30]. In eGeMAPS, 34 features are correlated above 80 percent; hence they are deleted, and the total features are 54.

With standardization, all the features will be transformed so that they will have the properties of a standard normal distribution with mean μ=0 and standard deviation σ=1. The formula for standardization is given as Eq. (2):

$$z = \frac{X - \mu}{\sigma} \quad (2)$$

An open-source, Java-based, WEKA data mining tool is used for performing normalization and standardization [31].

The dataset is split into train and development set, and testing set as 70%:30%. As the considered dataset is class imbalanced, the splitting is done concerning the class labels (70% of the data from class '0' and class '1' for training and validation, 30% of the data from class '0' and class '1' for testing). For model training and hyperparameters tuning the 70% of data is used, and 30% of the data is kept unseen for testing the model.

The training set of the DAIC-WOZ database contains 77 participants in the non-depressed class and 30 participants in the depressed class, which shows the severe class imbalance. Class imbalance in the training set leads to biased learning of the data model, which results in poor predictive accuracy over the minority classes and also suffers from underfitting. Underfitting is a scenario where the data model generates a high error rate on the training data and unseen test data. Hence, in this work ADASYN approach is used to overcome the class imbalance. ADASYN generates synthetic data for the minority class to bring up the class balancing.

3.3 Feature selection

The machine learning model makes predictions for the test data based on the training it underwent on the training data. However, it is obligatory to recognize which input data is adequate to training the model by removing the redundant data and irrelevant data. It reduces the dimension of the data and thus reduces the time complexity and improves the performance of the model [32]. This work uses the following methods for FS.

3.3.1 Boruta FS

This is a wrapper method built around the random forest classification algorithm. The core algorithm behind Boruta is random forests. Random forests are themselves based on decision trees. A decision tree is a sequence of steps (or decisions or splits) calculated at training time. A simple decision tree might classify every element as one of the two classes. Each new data point flows down the tree, either down the left or right branch, and ultimately arrives at its classification result.

A random forest, meanwhile, is an ensemble of weak decision trees. A random forest trains hundreds of purposefully over-fitted decision trees, with each decision tree only gaining access to a random subset of the columns in the dataset. To classify an incoming point, each of these trees cast a vote as to which class it assigns, and the majority vote wins. The key insight in random forests, and the reason that they perform better than decision trees alone, is mass voting. Increasing the randomness of the decision trees being built naturally increases their bias, but averaging their decisions, naturally reduces the variance. Suppose a random forest is able to decrease variance more than it increases bias, relative to a single well-pruned decision tree, it will perform better as a classifier on the dataset [33, 34].

Boruta technique takes this randomness even further. It seeks to capture all of the significant features in the dataset related to the target variable [35].

- It starts with adding randomness to the data by duplicating the dataset and rearranging the values in each column. These are known as shadow features. Following that, the dataset is used to train a classifier, such as RF Classifier. This makes sure that we can get a sense of the value of each feature in our data set using the Mean Decrease Accuracy or Mean Decrease Impurity. The better or more important the feature is, the higher the score.
- The algorithm then examines if the feature has a greater Z-score than its shadow features' maximum Z-score. If it does, it is stored in a vector, referred to as 'hit', and it will move on to the next iteration. After a predetermined number of iterations, it will generate a table of these 'hits'. A 'Z-score' is defined as the number of standard deviations a data point deviates from the mean.
- At each iteration, the algorithm compares the Z-scores of the shuffled copies of the features to the original features to see if the latter performed better. If it does, the feature is considered to be important by the algorithm. Essentially, the algorithm compares the feature's importance against randomly shuffled copies, increasing the technique's robustness. This is accomplished by utilizing a binomial distribution to compare the number of times each feature outperformed the shadow features.
- Constantly, it will reject a feature and remove it from the original matrix if it hasn't registered as a 'hit' in a predetermined number of iterations. After a specific number of iterations or after all of the features have been confirmed or rejected, it comes to an end.
- It uses an all-relevant FS strategy, which captures all features relevant to the outcome variable in some conditions. Most of the traditional FS algorithms, on the other hand, use a minimal optimum strategy in which they rely on a small group of features to provide a minimal error on a specified classifier.

In Boruta, all features strongly or weakly related to the decision variable are found, making it ideal for biological applications, such as determining which human genes (features) are linked to a specific medical problem (target variable).

3.3.2 Support vector machine-recursive feature elimination (SVM-RFE) FS

SVM-RFE is one of the successful FS approaches proposed by Singh et al. [36] in the selection of genes for cancer classification. It works by eliminating features recursively and constructing a model on the ones that remain. It takes two parameters into account: the estimator model to be used and the count of features to be selected. The accuracy measure is used to rank the features from most important to least important. It then ranks all the variables and provides support in the form of 'True' being significant and 'False' being irrelevant features. SVM-RFE works based on ranking the features using the equations discussed below.

Assume the training samples (x_1, y_1), (x_2, y_2), …, (x_l, y_l); $x_i \in R^n$; $y_i \in [-1,+1]$, with a target label y_i

SVM discrimination function is obtained by the Eq. (3).

$$f(x) = w^T x + b \tag{3}$$

where, x represents an input sample, b represents a bias, and w^T constitutes the weight vector acquired by the Eq. (4):

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{4}$$

where, α_i is the Lagrange multipliers or the support vectors.

After the SVM model has been developed, the weight vector can be determined on the trained model. The weight vector contains the weight assigned to every feature (input dimension), that indicates how important the features are for the classification process. In practice, the weight vector w^T, Eq. (4), which includes the feature values, is used in the training phase of the SVM model to assess features, and it removes a feature with a lower weight iteratively in the backward elimination process. SVM-RFE is a multivariate FS method that varies from other FS methods in that it relies on mutual information between features and target labels. It is better suited to data with varying noise fractions [37]. It is less prone to overfitting since it uses SVM to minimise structural risk in

statistical learning theory [38].

3.3.3 Fisher score-based FS

The Fisher score's primary principle is to select a subset of features that allows distances between data points in different classes to be as large as feasible in the data space covered by the selected features, while distances between data points in the same class to be as small as possible [39]. The process of calculating the fisher score of the features is presented as Algorithm1.

There are two labels, label1 is '0', which refers to a non-depressed participant, and label2 is '1' that refers to a depressed participant.

Algorithm 1: Pseudocode for Fisher score-based feature selection.
Input: training_data, number of features to be selected as '*n.*'
Output: Set of '*n*' features.
Method:
1: Read the training data set.
2: Calculate the mean and variance for label1 and label2, and also calculate the overall mean.
3: overall_mean=**mean**(training_data)
4: label1_mean= **mean**(label1)
5: label2_mean= **mean**(label2)
6: label1_variance=**variance**(label1)
7: label2_variance=**varance**(label2)
8: **for** each feature **do**
9: Calculate the fisher score
10: $numerator = (overall\_mean - label1\_mean)^2 + (overall\_mean - label2\_mean)^2$
11: $denominator = label1\_variance + label2\_variance$
12: $fisherscore\_value = \frac{numerator}{denominator}$
13: Sort the *fisherscore_value* in descending order to get the features with maximum fisher score on the top.
14: Store the sorted fisher score in *ranked_features*.
15: Select the required '*n*' number of features from the *ranked_features*
16: Extract the subset of training data concerning the selected '*n*' number of features and train the model.
17: Return '*n*' number of features

3.4 Classification

The following machine learning classifiers are used to classify people as depressed or non-depressed based on the characteristics of the data in terms of the selected suboptimal feature set.

3.4.1 Gaussian NB

Gaussian Naïve Bayes (GNB) is a Bayes theorem-based probabilistic classifier that assumes high (naïve) independence across the features. It is extremely linearly scalable with the number of features and data points, it is unaffected by irrelevant features, and can efficiently deal with the missing data. To determine conditional probability, the Bayes theorem can be used. It is used in machine learning since it is a valuable and effective tool in the study of probability. Bayes theorem formula is given as Eq. (5) and Eq. (6):

$$P(x|y) = \frac{P(x \cap y)}{P(y)} \tag{5}$$

$$P(x|y) = \frac{P(x).P(y|x)}{P(y)} \tag{6}$$

where, $P(x)$ is the probability of x occurring; $P(y)$ is the probability of y occurring; $P(x|y)$ is the probability of x given y; $P(y|x)$ is the probability of y given x; $P(x \cap y)$ is the probability of both x and y occurring.

One typical assumption when working with continuous data is that the continuous values associated with each class follow a normal (or Gaussian) distribution. Therefore the conditional probability of x,y variables is given by the Eq. (7):

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right) \tag{7}$$

where, μ_y represents the mean of y, and σ_y represents the standard deviation of y.

3.4.2 Support vector classifier

The support vector classifier (SVC) is a popular machine learning technique that performs exceptionally well on various classification challenges. Using the distance margin or distance between two support vectors, the SVC builds a hyperplane to split the dataset into numerous classes. As obtaining the best hyperplane in most cases necessitates data transfer to higher dimensions, various kernel functions are employed. The radial basis functions (RBF), polynomial, and linear are the three kernel functions used with SVM. The procedure for SVC is given below.

Using the Gaussian radial bias function kernel, as shown in Eq. (8), transform the original dataset into a higher-dimensional space.

$$K(X_i, Y_j) = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}} \tag{8}$$

Calculate the decision function as below:

Initially, construct the separating hyperplane using the Eq. (9):

$$S = x + \sum_{j=1}^{l} x_i y_i \tag{9}$$

where, y is an attribute value, l is the number of attributes, x is a scalar, and S is a separating hyperplane.

If $S>0$, the data point lies above the hyperplane S; If $S<0$, the data point lies below the hyperplane S.

Weights are adjusted to yield the hyperplane defining the sides of the margin.

$$M_1 = x_0 + \sum_{j=1}^{l} x_i y_i \geq 1 \text{ for } S_j = +1 \quad (10)$$

$$M_2 = x_0 + \sum_{j=1}^{l} x_i y_i \leq 1 \text{ for } S_j = -1 \quad (11)$$

If a data point fulfils the Eq. (12), then the support vectors are found.

$$S_j = \left(x_0 + \sum_{j=1}^{l} x_j y_j\right) \geq 1 \; \forall j \quad (12)$$

The maximum margin is calculated as $\frac{2}{\|x\|}$.
where,

$$\text{Euclidian norm} \|x\| = \sqrt{\sum_{j=1}^{l} x_i^2} \quad (13)$$

In order to obtain the hyperplane with maximum margin, the equation for S_j is transformed by using Lagrangian formulation and resolved by applying Karush Tucker conditions, also called first-order derivative tests.

The resulting decision boundary acquired is shown as Eq. (14):

$$Dec(Y^T) = \sum_{j=1}^{l} S_j b_j Y^j Y^T + x_0 \quad (14)$$

where, b, x_0 are the numeric parameters which are acquired from SVM optimization; Y^T is a testing sample, S_j is a class label of j^{th} sample.

If $Dec(Y^T)>0$ it is considered as a positive sample; else, it is considered as a negative sample. Finally, predict the class labels depending on the decision boundary.

3.4.3 K-Nearest neighbors (KNN)

The nearest neighbors (NN) classifiers, especially KNN classifiers, are simple yet efficient classification methods used widely in practice. It is a proven way of distinguishing between healthy and diseased people after selection of features, and also in bioinformatics [40]. The KNN rule uses a majority marking among its nearest neighbors to categorize each unknown instance in the given train set. The performance of this classifier is also highly influenced by the distance metric used to find the closest neighbors. If the previous information is not available, most of the KNN classifiers utilize simple Euclidean metrics to quantify the difference between samples denoted as vector inputs. The Euclidean distance between the samples is calculated using the Eq. (15):

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} w_r \left(a_r(x_i) - a_r(x_j)\right)^2} \quad (15)$$

where, an example is described as a vector $x=(a_1, a_2, \ldots, a_n)$, n is the count of samples in input, w_r is the weight of r^{th} attribute. Smaller the $d(x_i, x_j)$ more relevant are the samples. The test sample's class label is decided by the majority votes of its nearest K neighbors.

$$y(d_i) = \arg\max \sum_{x_j \in kNN} y(x_j, c_k) \quad (16)$$

where, d_i is an example of a test sample, x_j is one of the nearest neighbors to the training set, and $y(x_j, c_k)$ denotes whether x_j refers to the class c_k. Eq. (16) shows that the class with the majority of its members in K nearest neighbors will be chosen as the final predictor. It is worth noting that the K value plays a crucial role in model accuracy and computational expense. The smaller K value leads to having a higher noise effect on the result, and the larger K value increases the expense of computation. The plot between error rate and K, and the plot between accuracy and K should be checked before finalizing the K value. It is advisable to choose the K value, which gives the minimum error rate. Another simple approach to select the K value is $sqrt(n)$, where n is the samples count in training data.

3.4.4 Logistic regression

It is a classification technique that uses a sigmoid function to model the dichotomous dependent variable whose value lies in between [0, 1]. The sigmoid function used in LR is usually an S-shaped curve that emits one if the value is ≥0.5 and emits zero otherwise. The linear regression function computes the input to a sigmoid function. Using the cost function, the gradient descent approach is used to approximate the parameters of a linear function, such as weight and bias. The main outline of the principle of LR is given below.

Calculate the logistic regression function:

$$z = \beta_0 + \sum_{j=1}^{l} \beta_j y_j \quad (17)$$

where, l is the total number of attributes, β_0, β_j are scalar, and weight vector, and y_j represents the data sample.

We compute the Predictive probabilities using the Eq. (18):

$$p_{(\beta_0, \beta_j)}(z) = p_\theta(z) = \frac{1}{1+e^{-z}} \quad (18)$$

Using the cross-entropy, compute the cost function using the Eq. (19):

$$K(\beta_0, \beta_j) = K(\theta) = \frac{1}{L}\sum_{j=1}^{l} \left[m_j log\left(p_\theta(y_j)\right) + (1 - m_j) log\left(1 - p_\theta(y_j)\right)\right] \quad (19)$$

Using the gradient descent method, update the bias and weights.

$$\beta_j = \beta_j - \alpha d\beta_j \quad (20)$$

$$\beta_0 = \beta_0 - \alpha d\beta_0 \quad (21)$$

The estimated values β_0 and β_j are used for predicting the test data.

Then, compute the linear equation for the test data as shown in Eq. (22):

$$z = \beta_0 + \sum_{j=1}^{l} \beta_j y_j \quad (22)$$

Now, compute the probabilities using the sigmoid function:

$$p_{(\beta_0, \beta_j)}(z) = p_\theta(z) = \frac{1}{1+e^{-z}} \quad (23)$$

Finally, convert the probabilities into class labels using the decision boundaries as shown in Eq. (24):

$$\widehat{lab} = \begin{Bmatrix} 1; \; p_\theta(y) \geq 0.5 \\ 0; \; p_\theta(y) < 0.5 \end{Bmatrix} \quad (24)$$

where, $\widehat{lab}$ is the label or the class predicted.

3.4.5 Random Forest

Random Forest (RF) is a well-known supervised learning approach in machine learning. It can be used to solve problems involving classification and regression. It consists of numerous decision trees constructed on different subsets of the dataset. It follows the ensemble learning technique of combining different classifiers to solve a complex problem and enhance the performance of the model. Instead of relying on a single decision tree, it collects the decision from each constructed tree (binary tree). Depending on the majority votes of predictions, the final output is predicted. For each decision tree, the importance of the nodes is calculated using the Gini index as using the Eq. (25):

$$Gini = 1 - \sum_{i=1}^{c}(P_i)^2 \quad (25)$$

where, *P* represents the relative frequency of the class and *c* represents the count of classes.

Entropy can also be used to branch the nodes in a decision tree. The entropy is calculatedusing the Eq. (26):

$$Entropy = \sum_{i=1}^{c} -P_i * log_2(P_i) \quad (26)$$

here, also, *P* represents the relative frequency of the class, and *c* represents the count of classes. Because of the logarithmic function used to calculate entropy, it is mathematically more complex than the Gini index.

Classification problem that depends on the individual classifier for the entire dataset has the risk of misclassification and also low performance. Different classifiers make different errors on different samples; so, by combining classifiers, we can create an ensemble that produces more accurate predictions [41-43].

Therefore the researchers have come up with the Multiple Classifier System (MCS) approaches. Rather than a single classifier trained on entire data, multiple classifiers are trained and tested in MCS. In the end, the classifier with better performance is chosen. In recent times, various researches have published the demonstration and its pros over the individual classifiers [44, 45]. Several techniques for constructing an MCS are currently in use, and they have been presented in numerous outstanding reviews addressing various elements of MCS [46-48].

3.5 Dynamic ensemble selection of classifiers (DES)

Across a broad spectrum of classification issues, MCSs, that are made up of a pool of base classifiers perform better than their component classifiers [49]. Dynamic Selection (DS) is one of the best MCS techniques, where the base classifiers are chosen dynamically, based on each data sample to be classified. If more than one classifier is selected from a group of trained classifiers, it is referred to as Dynamic Ensemble Classification (DES). The design of MCS consists of three phases viz., over-production phase, training phase, and generalization phase. Figure 2 explains the DES workflow.

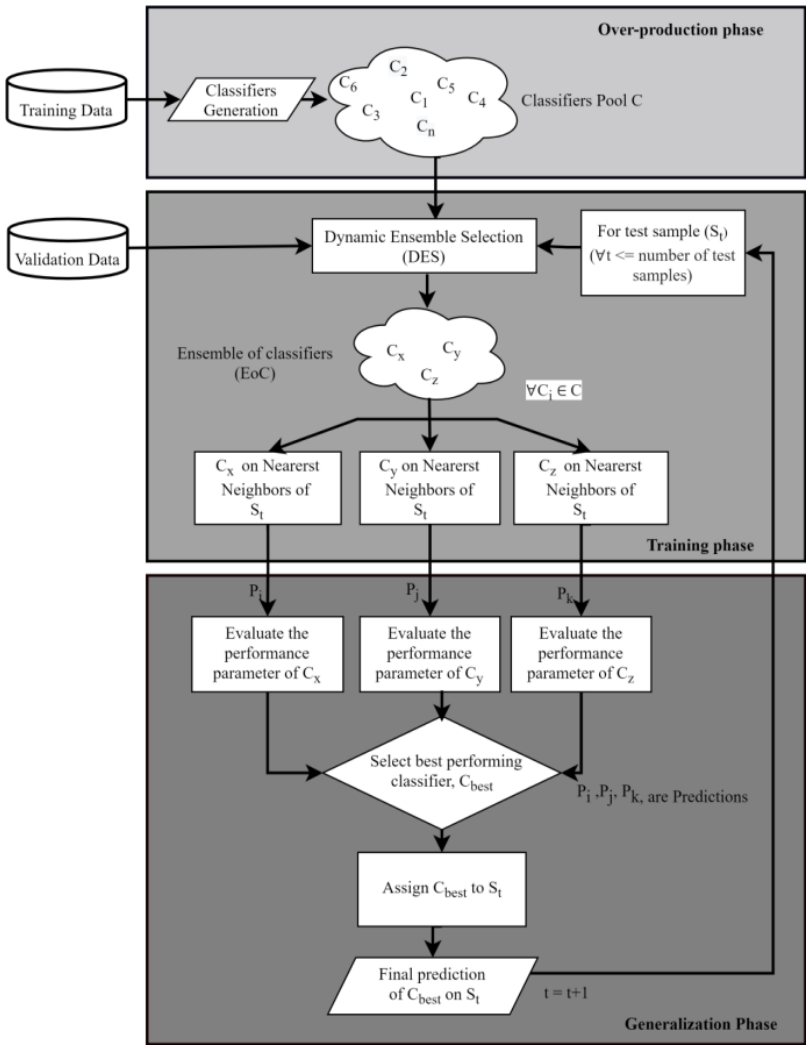

Figure 2. Dynamic ensemble selection of classifiers flow of execution

In the over-production phase, base classifiers' pool C is generated based on the training samples. These base classifiers can be of category homogenous classifies or heterogeneous classifiers. Homogeneous classifiers are trained by the same classifier on different datasets. The heterogeneous classifiers are obtained by applying different classifiers to the same training dataset. Some variation is expected in both scenarios. The work of Kuncheva and Whitaker [50] contains a thorough examination of several diversity measures. Researchers employed two sorts of approaches to create diverse classifiers: (1) implicit and (2) explicit [51, 52]. Unlike explicit approaches, implicit methods do not directly measure diversity. Different metrics for measuring diversity are proposed by Kuncheva and Whitaker [43].

Diversity is incorporated in homogenous pools by modifying the information used by adjusting the initial parameters, utilizing different subsets of the training data (Bagging [53], Boosting [54]). It is also incorporated by different classifier models and their combinations like KNN, SVM, and decision tree classifiers or using various feature subspaces (Random Subspace Selection [55]).

The training phase includes the selection of either a single classifier C_i or a subset C` of classifiers $C` \subset C$ with respect to the notion of classifier competence on the single test data S_t. If multiple classifiers are selected, it is called as Ensemble of Classifiers (EoC). The accuracy of the classifiers in the local area is one of the selection criteria. The accuracy of the classifiers in the local area is one of the selection criteria [56, 57], and other criteria involve ranking [58], probabilistic model [59], and oracle information [60, 61]. Assume that the selected EoC has C_x, C_y, and C_z, $\forall C_i \in C`$, then these classifiers are trained on the K nearest neighbors of the test data sample S_t. The default value of K is 7, which can be changed as per the experimentation. In the generalization phase, each classifier's performance is checked with its performance evaluation parameter and found out which classifier (C_{best}) has shown the best performance on the nearest neighbors of S_t. Finally, the C_{best} classifier is assigned to the test data sample S_t and the prediction of C_{best} (S_t) is obtained. The second and third phases are repeated for all the samples of the test data. The KNORA-E, KNORA-U, METADES, and DES-MI dynamic ensembles are explored for this work.

3.5.1 KNORA-eliminate (KNORA-E)

The KNORA-E method is an Oracle-based approach [62] based on the performance of the classifier in a local region specified by the K-nearest neighbors in the validation set of the test sample to be classified. KNORA-E selects the classifiers that categorize all K nearest neighbors correctly. If there is no such classifier present, the K value is reduced by one, and the process is repeated.

Given K neighbors of a test sample S_t and assume that the ensemble of classifiers $C`$ classifies all of its $K$-nearest neighbors correctly. Each classifier $C_i \in C`$ that belongs to the $C`$ shall submit a vote on that particular test sample S_t. If there is no classifier that can correctly categorize all K-nearest neighbors of S_t, then the value of K is reduced until at least one classifier correctly classifies all of the neighbors [61]. The procedure of the KNORA-E model is given in Algorithm 2. The selected classifiers are combined by the majority voting method.

Algorithm 2: KNORAE
Input: Classifiers pool, *C;* validation set S_{val}; testing sample S_t; nearest neighborhood size K.
Output: Ensemble of classifiers C`(S_t).
Method:
for every test sample S_t in Test Data **do**
  k=K;
  while k>0 **do**
    Find Φ as the K-nearest neighbors of S_t in validation set S_{val}
    for each classifier C_i in C **do**
      if (C_i recognizes all sample correctly in Φ) **then**
        C`=C` ∪ C_i;
      end if
    end for
      if(C`==∅) **then**
        K=K-1
      else
        break;
      end if
  end while
      if(C`==∅) **then**
        find out the classifier C_i which recognizes most of the samples in Φ correctly;
        Choose the classifiers that can recognize the same number of samples of C_i to construct the ensemble C`;
      end if
        Use the final ensemble C` for classifying S_t;
end for

3.5.2 KNORA-Union (KNORA-U)

This method, unlike KNORA-E, selects the classifier if it classifies correctly at least one of the K nearest neighbors of test sample S_t [63]. Given K neighbors, S_t, a test sample of test data, assume that the j-nearest neighbor, $1 \leq j \leq K$, is correctly classified by a set of classifiers C`, then each classifier $C_i \in C`$ shall offer a vote on the sample S_t. It's worth noting that, as all K-nearest neighbors are taken into account, a classifier can receive multiple votes if it properly classifies more than one nearest neighbor. If more neighbors are classified correctly, the classifier gets more votes for a test set. The final result is obtained using the weighted majority voting method. Algorithm 3 portrays the KNORA-U model.

Algorithm 3: KNORAU
Input: Classifiers pool, *C;* validation set S_{val}; testing sample S_t; nearest neighborhood size *K*.
Output: Ensemble of classifiers $C`(S_{t)}$.
Method:
1: **for** every test sample S_t in test data **do**
2: k=K;
3: **while** k>0 **do**
4: Find Φ as the K-nearest neighbors of S_t in validation set S_{val}
5: **for** every sample Φ_i in Φ **do**
6: **for** every classifier C_i in C **do**
7: **if**(Ci classify Φ_i correctly) **then**
8: $C` = C` \cup C_i$
9: **end if**
10: **end for**
11: **end for**
12: Use the final ensemble C` for classifying S_t.
13: **end while**
14: **end for**

3.5.3 META-DES

The META-DES framework assumes that the DES problem can be considered a meta-problem. The meta-problem is a set of problems that make up a single problem. To decide whether or not a base classifier C_i is competent enough for classifying a given test sample, this meta-problem employs a variety of criteria pertaining to its behavior. It follows the meta-learning framework that contains meta-problem, meta-features, and meta-classifier [64-67].

The meta-problem is to determine whether a base classifier C_i is capable of classifying S_t. Each meta-feature F_m corresponds to a separate criterion for assessing the base classifier's competency. The meta-features are stored into a vector of meta-features $V_{i,j,}$ that has the details of how a base classifier C_i behaves in relation to the input instance S_t. Depending on the meta-features vector $V_{i,j}$, a meta-classifier μ is trained to predict whether C_i will make a proper prediction for S_t. That is, based on $V_{i,j}$, a meta-classifier μ is trained to predict if a base classifier C_i is capable of classifying given a test sample S_t. This work uses the default meta-classifier that is a multinomial naïve Bayes for the experimentation. As a result, the proposed method varies from current state-of-the-art dynamic selection strategies not only in that it employs several criteria but also in that the selection rule uses the meta-classifier μ for learning using the training data. The final decisions of base classifiers in C` are combined by the weighted majority voting method. The steps involved in obtaining ensemble of classifiers in META-DES are explained in Algorithm 4. As it has been effectively employed by other DES approaches, the majority vote method is used to integrate the selected classifiers [62].

Algorithm 4: META-DES
Input: Classifiers pool, C; Test sample S_t; dynamic selection dataset $D_{sel.}$
Output: Ensemble of classifiers $C`(S_{t)}$.
1: C`=∅
2: Discover the region of competence Θ_t of S_t using $D_{sel.}$
3: Calculate the output profile $\tilde{S}_t$ of $S_{t.}$
4: Find the K_p similar output profiles Φ_t of $\tilde{S}_t$ using $\tilde{D}_{sel}$.
5: **for** all $C_i \in C$ **do**
6: $V_{i,j}$=FeatureExtraction($\Theta_{t,}$ $\Phi_{t,}$ C_i, S_t)
7: input $V_{i,j}$ to μ
8: **if** the class attribute $\alpha_{i,t}$=1 "C_i is competent for S_t" **then**
9: C`=C`∪{C_i}
10: **end if**
11: Use the final ensemble C` for classifying S_t;
12: **end for**

3.5.4 DES for multi-class imbalanced datasets (DES-MI)

Several real-world problems of classification undergo the problem of class imbalance, where some classes are significantly underrepresented compared to others. As a solution, García et al. [68] devised DESMI, an innovative and successful method for evaluating candidate classifiers' ability using weighted instances in the neighborhood. In this method, the generating the balanced training datasets and the selection of suitable classifiers are two crucial components. The random balance framework achieves the diversity of classifiers in the candidate pool by combining the approaches of random under-sampling (RUS), random over-sampling (ROS), and synthetic minority oversampling technique (SMOTE) [69]. Then, using the weighted instances in the neighborhood and the test sample S_t, the competency of candidate classifiers is assessed. It considers higher competence in a classifier that is more effective in classifying minority classes in the local area. Finally, each chosen classifier casts a vote on the test sample S_t. The votes casted for each class are totaled, and the class with the most votes is selected as the final output class. Algorithm 5 briefs the procedure of choosing the ensemble of classifiers using DES-MI.

Algorithm 5: DES-MI

Input: Base classifier pool C, base learner y, Training dataset D_{tr}, testing dataset D_{test}, validation dataset D_{val}, number of nearest neighbors K, percentage of classifiers to be selected P%, the scaling coefficient α

Output: Ensemble of classifiers C`(St).

£ ← ∅
Generate candidate classifier pool £
    Generate balanced training dataset D_{tr} by producing synthetic data samples of a minority class like ADASYN method
    Create
        h_t ← y(D_{tr}) //competent base classifier h_t
        £ ← £ ∪ h_t
for every testing sample S_t in D_{test}, **do**
    C` ← ∅
    find Θ as K nearest neighbors of S_t in D_{val}
    for each sample x_i in Θ **do**
        num ← **count** the number of samples with the same class as x_i
        $w_i \leftarrow \frac{1}{1+\exp(\alpha * num)}$ //calculate the voting weights for each x_i in Θ
    end for
    Normalize w_i according to $\widehat{w} \leftarrow \frac{w_i}{\sum_{i=1}^{k} w_i}$
    for each classifier h_t in £ **do**
        $\acute{C}(h_t|S_t) \leftarrow \sum_{i=1}^{k} I(h_t(S_t) = y_t) * \widehat{w}$ //y_t is the class label of S_t
    end for
    select P% most competent classifiers in £ to compose the ensemble C` for test sample S_t.
end for

3.5.5 Repeated stratified K-Fold cross-validation

Cross-validation (CV) is a process of assessing a machine learning model using data resampling when there is a small dataset. Data resampling is a technique in which the data samples are repeatedly drawn from a dataset and used for training and testing the model. *K*-Fold is based on random sampling where data is split into *K* number of disjoint blocks (the folds), which are approximately equal, by choosing the samples randomly. Repeatedly, from the *K* number of folds, each fold is used for testing once, and the remaining *K-1* folds are used for training the model. The model is executed *K* number of times; hence, the model's performance is the mean of the model's performance on each fold.

Random sampling does not perform well in the case of an imbalanced dataset as there is a chance of choosing more samples of one class than the other class, which may lead to biased predictions. To mitigate this problem, this work uses the *Stratified K-Fold* technique for CV [70]. *Stratified K-Fold* is a variation of the *K-Fold* CV, which uses stratified sampling. In stratified sampling, the percentage of samples of each class is preserved in each fold while dividing the data into folds. Further, this entire process can also be repeated multiple times; for this work, we have chosen the number of folds as 10 and the number of repetitions as 2.

3.5.6 Performance metrics

The performance of any classifier is evaluated by using performance metrics. The output of any individual machine learning classifier or dynamic ensemble of classifiers is interpreted from the confusion matrix that it generates. A confusion matrix is the summary of predictions of any classifier on a set of test data. In this work, the classification is of a type binary, which means that every participant from the test set is predicted to be either in a depressed class (denoted as 1) or a non-depressed class (denoted as 0). A confusion matrix contains four basic terms, as given below, which can be used to calculate different performance metrics.

True Positives (TP): A participant is actually 'depressed,' and the prediction is also 'depressed'.

True Negatives (TN): A participant is actually 'non-depressed,' and the prediction is also 'non-depressed'.

False Positives (FP): A participant is 'non-depressed', but the classifier is predicted as 'depressed'. (It is also called a "Type I error").

False Negatives (FN): A participant is actually 'depressed,' but the classifier predicted as 'non-depressed'. (It is also called a "Type II error"). Performance metrics are calculated as shown in Table 3.

Table 3. Metrics used in this work

Metric	Formula
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Recall/Sensitivity	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
Specificity	$\frac{TN}{TN+FP}$
Balanced Accuracy	$\frac{Sensitivity+Specificity}{2}$
F-measure	$2 * \frac{Precision*Recall}{Precision+Recall}$

Note 1: TP: True positives; TN: True Negatives; FP: False Positives; FN: False Negatives

Accuracy is the measurement of the correctness of the classifier's predictions. Sensitivity, also called Recall, refers to the ability of a classifier to designate the depressed participant as 'depressed'. A highly sensitive classifier means that there are very few 'False Negative' predictions. Precision denotes the confidence of the classifier in predicting the class labels. In this work, precision is the measure of 'depressed' that the classifier correctly identifies out of all the depressed participants. Specificity is the ability of a classifier to classify the participant who is non-depressed as 'non-depressed'. A highly specific classification means that there are very few

'False Positives' in the resulting predictions. Balanced accuracy, also called Binary Classification Accuracy (BCA), is an essential metric for a binary classifier when the dataset is imbalanced, i.e., when one of the target classes appears a lot more than the other. The dataset that this work uses has class imbalance; hence, the balanced accuracy is also presented. The final metric is F-measure which is the harmonic mean of the precision and recall giving each the same weightage. It provides a better measure of the wrongly predicted cases than the accuracy metric.

4. RESULTS AND DISCUSSION

To perform the experiments, we used Intel(R) Core™ i5-9300H CPU @ 2.40GHz acer Aspire 7 machine. The proposed model starts with FS to obtain a suboptimal feature set that has high importance. We have utilized the GridSearchCV technique for hyperparameter tuning. For cross-validation, the Stratified 10-fold method is used with the number of repetitions as 2. From the recent researches in applied machine learning, particularly in the healthcare domain, a 10-fold CV fetched good performance [71-74]. The performance metrics are calculated for all the10-folds of the evaluation and averaged to get the final metrics.

4.1 Results with Boruta FS

Boruta selects only 10 features as the features of high importance from the total 54 features. They are F0semitoneFrom27.5Hz_sma3nz_pctlrange02,slopeUV0500_sma3nz_amean, F1amplitudeLogRelF0_sma3nz_amean, F1amplitudeLogRelF0_sma3nz_stddevNorm,F2bandwidth_sma3nz_stddevNorm,mfcc1_sma3_amean,mfcc3V_sma3nz_stddevNorm,F3bandwidth_sma3nz_amean,slopeV0500_sma3nz_stddevNorm, mfcc4_sma3_amean. Table 4 lists the selected features using Boruta FS.

Table 4. Features selected using Boruta FS

S. No.	Selected features
1	F0semitoneFrom27.5Hz_sma3nz_pctlrange0-2
2	slopeUV0-500_sma3nz_amean
3	F1amplitudeLogRelF0_sma3nz_amean
4	F1amplitudeLogRelF0_sma3nz_stddevNorm
5	F2bandwidth_sma3nz_stddevNorm
6	mfcc1_sma3_amean
7	mfcc3V_sma3nz_stddevNorm
8	F3bandwidth_sma3nz_amean
9	slopeV0-500_sma3nz_stddevNorm
10	mfcc4_sma3_amean

The performance of base classifiers is evaluated using the metrics on the selected features, and the results are tabled from Table 5 to Table 9 for the selected 5 and 10 features. All the classifiers are operating on the tuned hyperparameters for optimal performance.

The base classifier SVC shows the accuracy as 74% and balanced accuracy as 73% when experimented with 10 features. Precision and sensitivity are relatively offering poor performance. The graph showing all the base classifiers' performance on 5 features and 10 features are depicted in Figure 3 and Figure 4, respectively.

The DES classifiers' performance summary on the features selected with boruta FS for 5 and 10 features is presented in Table 10. KNORA-U, when k=3, has shown a bit improved performance than the individual base classifier. It has given the accuracy as 77% and balanced accuracy as 75% for the selected subset of 10 features. One more DES method META-DES has also shown accuracy similar to KNORA-U, but the balanced accuracy is low, which is 69%.

The accuracy and balanced accuracy of all the four DES classifiers on a subset of 5 features as well as a subset of 10 features are shown graphically in Figure 5 and Figure 6. It is known from the graph that KNORA-U DES classifier with k=3 has given accuracy as 77% and balanced accuracy as 75% using the subset 10 optimal features.

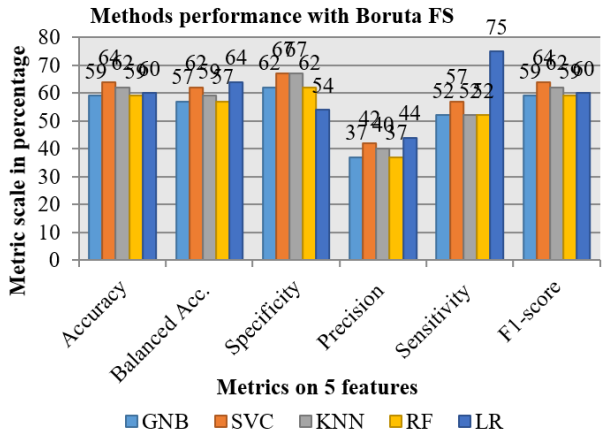

Figure 3. Base classifiers performance on 5 Boruta selected features

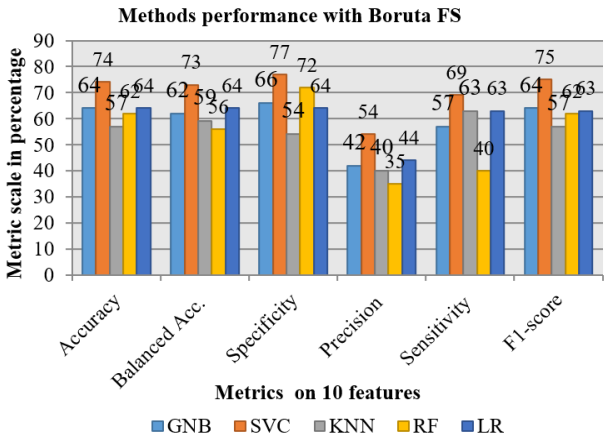

Figure 4. Base classifiers performance on 10 Boruta selected features

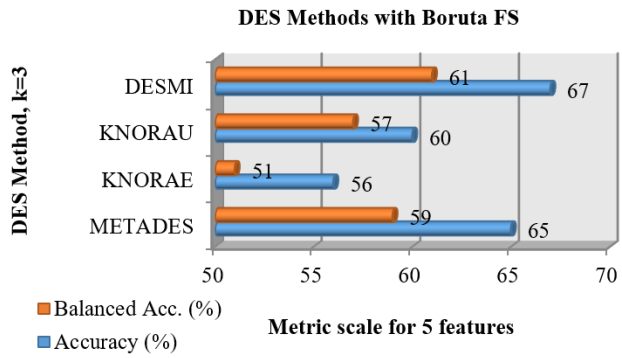

Figure 5. DES classifiers performance on 5 Boruta selected features

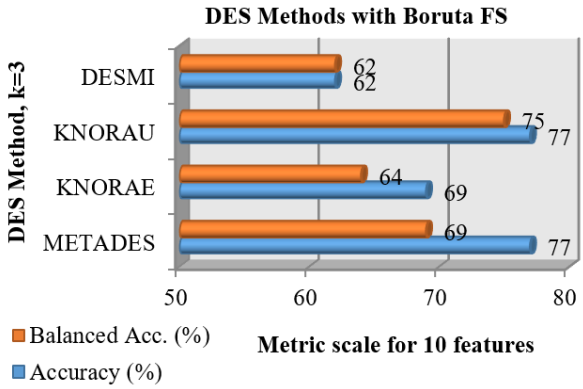

Figure 6. DES classifiers performance on 10 Boruta selected features

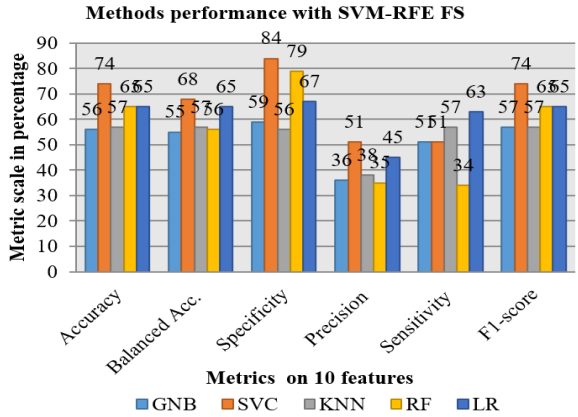

Figure 7. Base classifiers performance on 10 SVM RFE selected features

4.2 Results with SVM-RFE FS

SVM-RFE has given 25 features as features of high importance, as shown in Table 11. It is experimented with by dividing these features into different feature subsets of 5, 10, 15, and 20 features. Initially, the individual base classifiers are checked against these feature subsets and tabulated the results from Table 12 to Table 16.

SVC has shown better performance with the feature subsets of 20 and 10 features. The SVC using the feature subset of 20 features has given the accuracy as 76% and the balanced accuracy as 69%. SVC, utilizing the feature subset of 10 features also, has provided relatively good accuracy, 74%, and balanced accuracy of 68%. Performance of the entire individual base classifiers is shown in Figure 7 with the feature subset of 10 features and Figure 8 with the feature subset of 20 features.

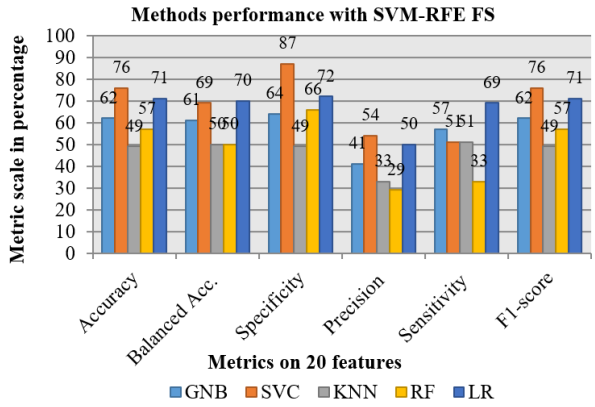

Figure 8. Base classifiers performance on 20 SVM RFE selected features

Table 5. Gaussian NB performance on Boruta FS features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	59	57	62	37	52	59
10	64	62	66	42	57	64

Table 6. SVC performance on Boruta FS features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	64	62	67	42	57	64
10	74	73	77	54	69	75

Table 7. KNN performance on Boruta FS features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	62	59	67	40	52	62
10	57	59	54	40	63	57

Table 8. RF classifier performance on Boruta FS features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	**59**	**57**	**62**	**37**	**52**	**59**
10	**62**	**56**	**72**	**35**	**40**	**62**

Table 9. LR performance on Boruta FS features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	60	64	54	44	75	60
10	64	64	64	44	63	63

Table 10. Summary of DES classifiers performance on Boruta FS features with RepeatedStratified 10-fold CV

DES Method	# of Features	Acc. (%)	Bal. Acc. (%)	Spec. (%)	Prec. (%)	Sens. (%)	F1-score (%)
META DES K=3	5	65	59	74	39	45	65
	10	77	69	88	48	48	76
META DES K=5	5	62	55	72	35	39	62
	10	67	63	74	43	51	67
KNORAE K=3	5	56	51	64	32	39	56
	10	69	64	77	44	50	69
KNORAE K=5	5	58	56	61	38	50	58
	10	65	63	69	43	57	65
KNORAU K=3	5	60	57	64	38	51	60
	10	77	75	77	55	72	76
KNORAU K=5	5	62	58	66	39	50	62
	10	72	70	77	50	62	72
DESMI K=3	5	67	61	77	40	45	67
	10	62	62	61	43	62	62
DESMI K=5	5	69	64	77	44	50	69
	10	67	64	72	44	56	67

Table 11. Features selected using SVM-RFE FS

S. No.	Selected features	S. No.	
1	F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope	14	mfcc3_sma3_stddevNorm
2	F1amplitudeLogRelF0_sma3nz_amean	15	jitterLocal_sma3nz_amean
3	F3bandwidth_sma3nz_amean	16	mfcc4_sma3_amean
4	F3bandwidth_sma3nz_stddevNorm	17	shimmerLocaldB_sma3nz_amean
5	slopeUV0-500_sma3nz_amean	18	HNRdBACF_sma3nz_stddevNorm
6	mfcc1_sma3_stddevNorm	19	mfcc1V_sma3nz_amean
7	F1amplitudeLogRelF0_sma3nz_stddevNorm	20	mfcc3V_sma3nz_stddevNorm
8	F2bandwidth_sma3nz_stddevNorm	21	F0semitoneFrom27.5Hz_sma3nz_meanRisingSlope
9	slopeV500-1500_sma3nz_amean	22	mfcc2_sma3_stddevNorm
10	mfcc4V_sma3nz_stddevNorm	23	mfcc3_sma3_amean
11	F0semitoneFrom27.5Hz_sma3nz_amean	24	logRelF0-H1-H2_sma3nz_amean
12	loudness_sma3_stddevNorm	25	slopeV500-1500_sma3nz_stddevNorm
13	loudness_sma3_stddevRisingSlope		

Table 12. GNB performance on SVM-RFE features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	64	58	72	39	45	64
10	56	55	59	36	51	57
15	53	51	56	33	45	53
20	62	61	64	41	57	62
25	58	57	59	39	57	58

Table 13. SVC performance on SVM-RFE features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	64	55	77	34	34	64
10	74	68	84	51	51	74
15	65	63	69	43	57	65
20	76	69	87	54	51	76
25	73	65	84	48	45	73

Table 14. KNN performance on SVM-RFE features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	64	65	61	45	69	64
10	57	57	56	38	57	57
15	55	55	54	37	57	55
20	49	50	49	33	51	49
25	48	53	38	36	69	48

Table 15. RF performance on SVM-RFE features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	64	59	72	39	45	64
10	65	56	79	35	34	65
15	58	50	72	27	28	58
20	57	50	66	29	33	57
25	65	56	79	35	33	65

Table 16. LR performance on SVM-RFE features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	67	69	64	48	75	67
10	65	65	67	45	63	65
15	69	66	74	46	57	69
20	71	70	72	50	69	71
25	65	66	64	46	69	66

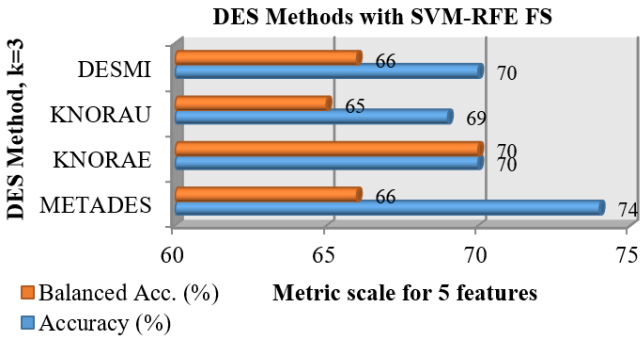

Figure 9. DES classifiers performance on 5 SVM-RFE selected features

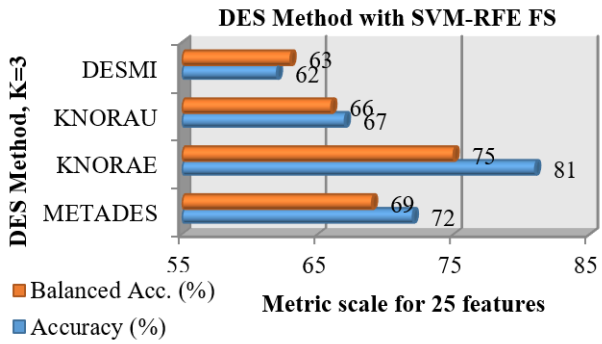

Figure 10. DES classifiers performance on 25 SVM-RFE selected features

Table 17. Summary of DES classifiers performance on SVM-RFE features with RepeatedStratified 10-fold CV

DES Method	# of Features	Acc. (%)	Bal. Acc. (%)	Spec. (%)	Prec. (%)	Sens. (%)	F1-score (%)
METADES K=3	5	74	66	87	46	45	74
	10	67	61	77	40	45	67
	15	60	57	64	38	50	60
	20	65	61	72	41	50	65
	25	72	69	77	50	62	72
METADES K=5	5	72	64	84	44	44	72
	10	70	65	79	45	50	70
	15	67	62	74	42	50	67
	20	69	65	74	46	56	69
	25	63	61	66	42	56	63
KNORAE K=3	5	70	70	71	50	68	71
	10	65	61	72	41	50	65
	15	60	57	64	38	50	60
	20	65	61	72	41	50	65
	25	81	75	90	57	60	81
KNORAE K=5	5	63	60	69	40	50	63
	10	65	63	69	43	56	65
	15	63	65	61	45	68	63
	20	70	67	77	47	56	71
	25	67	62	74	42	50	67
KNORAU K=3	5	69	65	74	45	56	69
	10	65	63	69	43	56	65
	15	67	64	71	44	56	67
	20	67	64	72	44	56	67
	25	67	66	69	46	62	67
KNORAU K=5	5	63	61	66	42	56	63
	10	69	65	74	46	56	69
	15	60	59	61	40	56	60
	20	69	67	72	47	62	69
	25	69	65	74	46	56	69
DESMI K=3	5	70	66	77	47	56	70
	10	60	57	64	38	50	60
	15	60	59	61	40	56	60
	20	63	63	64	44	62	63
	25	62	63	59	44	68	61
DESMI K=5	5	74	68	84	49	50	74
	10	58	56	61	38	50	58
	15	58	57	59	39	56	58
	20	69	67	72	47	62	69
	25	62	63	59	44	68	62

The DES classifiers have shown improved results than the individual base classifiers. The summary results of DES are shown in Table 17. KNORA-E with *k*=3 on the feature subset of 25 features has demonstrated the accuracy of 81% and balanced accuracy as 75%. META-DES *k*=3 and DES-MI *k*=5 have shown similar accuracy of 74% on the feature subset of 5 features.

The graphs showing the accuracy and balanced accuracy of DES classifiers on feature subsets of 5 and 25 features are depicted in Figure 9 and Figure 10, respectively.

4.2 Results with fisher score-based FS

Table 18 presents the features of high importance based on their fisher score. The features designated as the features of high importance have the fisher score ranging from 0.4596 to 3.4548. By observing the experimentation results of classifiers on the features selected based on fisher score, it is evident that these features have shown superior performance compared to the Boruta FS and SVM-RFE FS. We have experimented with different feature subsets with 5, 10, 15, 20, and 25 features. All the feature subsets with individual base classifiers are tabulated from Table 19 to Table 23.

In the case of individual base classifiers, performance is shown graphically for a feature subset exhibiting a potentially improved performance. Performance in graphs is shown in Figure 11 with 5 features, Figure 12 with 20 features, and Figure 13 with 10 features. Both the base classifiers KNN and GNB have demonstrated an accuracy of 69% for the subset of 5 and 20 features. Concerning balanced accuracy, KNN has given a good result with 67% on the subset of 5 features.

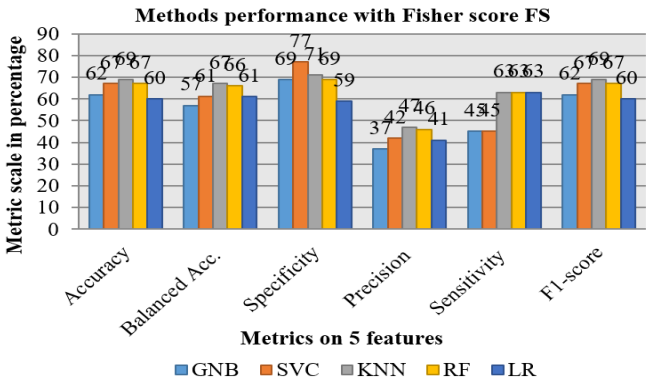

Figure 11. Base classifiers performance on 5 fisher score selected features

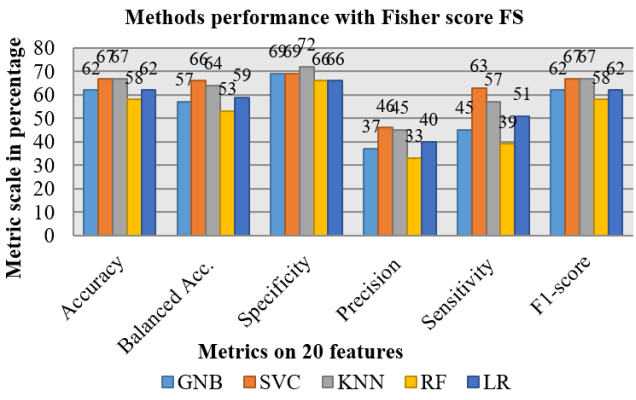

Figure 12. Base classifiers performance on 20 fisher score selected features

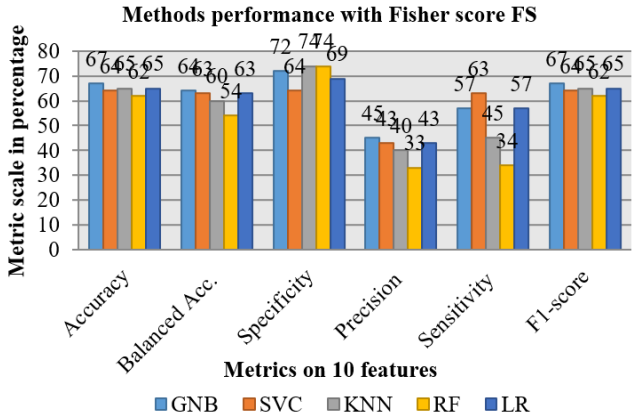

Figure 13. Base classifiers performance on 10 fisher score selected features

DES classifiers have improved performance using the feature subsets selected using Fisher score-based FS. KNORA-U with *k*=5 has shown superior performance of 82% accuracy and 77% balanced accuracy for the feature subset of 15 features that outperforms state-of-the-art models using DAIC-WOZ for depression detection.

The next best performing DES classifier is also KNORA-U with *k*=3 on a feature subset of 20 features with an accuracy of 81% and balanced accuracy of 75%. KNORA-U shows improved performance than other DES classifiers as it chooses classifiers from the pool of base classifiers, as Ensemble of classifiers, that accurately classify at least one sample from the query sample's region of competence.

The summary performance of DES classifiers is shown in Table 24 for feature subsets of different sizes. The graphs of DES classifiers with varying subsets of features with 15 and 20 features with improved performance are shown in Figure 14 and Figure 15, respectively.

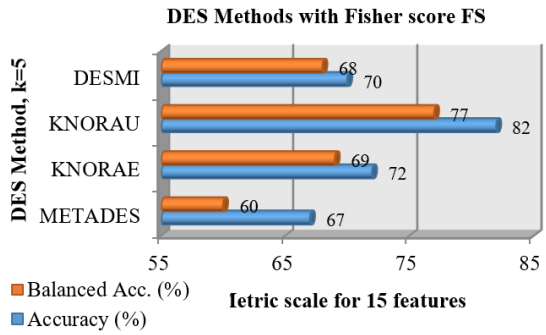

Figure 14. DES classifiers performance on 15 fisher score features

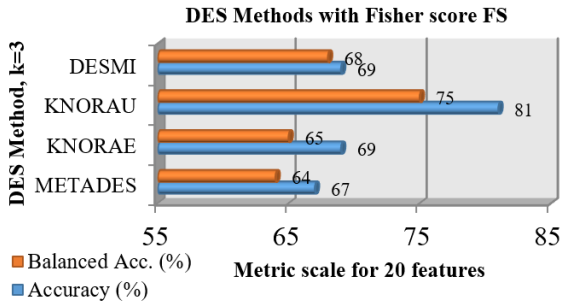

Figure 15. DES classifiers performance on 20 fisher score features

Table 18. Selected features based on fisher score

S. No.	Features selected	Fisher score	S. No.	Features selected	Fisher score
1	F2bandwidth_sma3nz_amean	3.4548	11	mfcc4_sma3_amean	2.1932
2	F1amplitudeLogRelF0_sma3nz_amean	3.4212	12	F0semitoneFrom27.5Hz_sma3nz_meanRisingSlope	1.8869
3	F1bandwidth_sma3nz_stddevNorm	3.3575	13	mfcc3_sma3_amean	1.4009
4	F2amplitudeLogRelF0_sma3nz_amean	3.3467	14	F0semitoneFrom27.5Hz_sma3nz_stddevRisingSlope	1.1046
5	F1amplitudeLogRelF0_sma3nz_stddevNorm	3.3309	15	mfcc4_sma3_stddevNorm	1.045
6	F2amplitudeLogRelF0_sma3nz_stddevNorm	3.2543	16	loudness_sma3_stddevRisingSlope	0.8519
7	mfcc2_sma3_amean	3.1756	17	logRelF0-H1-A3_sma3nz_amean	0.8343
8	F2bandwidth_sma3nz_stddevNorm	2.9340	18	spectralFlux_sma3_stddevNorm	0.6793
9	F0semitoneFrom27.5Hz_sma3nz_stddevFallingSlope	2.6645	19	loudness_sma3_pctlrange0-2	0.4833
10	F1bandwidth_sma3nz_amean	2.4462	20	logRelF0-H1-H2_sma3nz_stddevNorm	0.4596

Table 19. Gaussian NB performance on fisher score-based features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	62	57	69	37	45	62
10	67	64	72	45	57	67
15	57	53	61	34	45	56
20	69	66	74	46	57	69
25	62	63	61	42	63	62

Table 20. SVC performance on fisher score-based features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	67	61	77	42	45	67
10	64	63	64	43	63	64
15	65	60	74	40	45	65
20	67	66	69	46	63	67
25	65	58	76	38	39	65

Table 21. KNN performance on fisher score-based features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	69	67	71	47	63	69
10	65	60	74	40	45	65
15	60	59	41	63	60	60
20	67	64	72	45	57	67
25	55	54	56	35	51	55

Table 22. RF performance on fisher score-based features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	67	66	69	46	63	67
10	62	54	74	33	34	62
15	64	55	67	34	34	64
20	58	53	66	33	39	58
25	56	48	69	26	28	57

Table 23. LR performance on fisher score based features with RepeatedStratified 10-fold CV

# of Features	Accu. (%)	Bal. Acc. (%)	Speci. (%)	Prec. (%)	Sens. (%)	F1-score (%)
5	60	61	59	41	63	60
10	65	63	69	43	57	65
15	60	57	64	38	51	60
20	62	59	66	40	51	62
25	65	58	77	38	40	65

Table 24. Summary of DES classifiers performance on fisher score based features with RepeatedStratified 10-fold CV

DES Method	# of Features	Acc. (%)	Bal. Acc. (%)	Spec. (%)	Prec. (%)	Sens. (%)	F1-score (%)
METADES K=3	5	63	56	74	35	39	63
	10	**74**	67	85	49	50	74
	15	69	67	73	47	62	69
	20	67	64	72	44	56	67
	25	72	66	82	47	50	72

	5	70	65	79	45	50	70
	10	72	68	79	49	56	72
METADES K=5	15	67	60	76	40	45	67
	20	70	65	79	45	50	70
	25	72	68	79	48	56	72
	5	70	65	79	45	50	70
	10	74	67	84	48	50	74
KNORAE K=3	15	70	68	74	48	62	70
	20	69	65	74	45	56	69
	25	**76**	70	84	52	56	76
	5	72	68	79	48	56	72
	10	72	69	77	50	62	72
KNORAE K=5	15	72	69	77	50	62	72
	20	72	66	82	47	50	72
	25	72	66	82	47	50	72
	5	72	68	79	48	56	72
	10	74	69	82	50	56	74
KNORAU K=3	15	75	71	79	51	62	75
	20	**81**	75	90	57	59	80
	25	69	63	77	44	50	69
	5	72	68	79	48	56	72
	10	72	68	79	48	56	72
KNORAU K=5	15	**82**	77	88	59	66	82
	20	76	69	87	50	50	76
	25	72	69	77	50	62	72
	5	72	68	79	48	56	72
	10	70	65	79	45	50	70
DESMI K=3	15	69	65	74	45	56	69
	20	69	68	69	48	68	69
	25	70	70	71	50	68	70
	5	76	70	84	52	56	76
	10	70	63	82	43	44	70
DESMI K=5	15	70	68	74	48	62	70
	20	69	67	71	47	62	69
	25	70	70	71	50	68	70

5. CONCLUSION AND FUTURE WORK

A mental disorder, depression is prevailing rapidly worldwide, and most of the cases have been identified in the last stage of the disease. Researchers have found discrimination in the speech of a depressed and non-depressed person. In this paper, we extracted the acoustic characteristics of a person's speech recording and used base classifiers as GNB, SVC, KNN, RF, and LR to predict whether the person is depressed or not. Our main objective is to find the sub-optimal feature set that can effectively predict the depression and a DES classifier to improve the prediction accuracy. Therefore we have employed Boruta, SVM-RFE, and Fisher score-based FS techniques and META-DES, KNORA-E, KNORA-U, and DES-MI DES classifiers. The experimentations are performed on the publicly available and clinically validated DAIC-WOZ dataset. Our model has shown that the KNORA-U DES classifier with the five base classifiers pool gives improved performance. It has given the accuracy and balanced accuracy as 82% and 77%, respectively, when the suboptimal feature set of 15 features of the fisher score-based FS is used. Our findings also show that the DES classifiers can improve the predictions compared to the individual base classifiers. In the future, we would like to work on developing an efficient feature selection method to improve the accuracy further and also work on creating a primary dataset.

REFERENCES

[1] Hidaka, B.H. (2012). Depression as a disease of modernity: Explanations for increasing prevalence. Journal of Affective Disorders, 140(3): 205-214. https://doi.org/10.1016/j.jad.2011.12.036

[2] Marcus, M., Yasamy, M.T., van Ommeren, M.V., Chisholm, D., Saxena, S. (2012). Depression: A global public health concern. Available: https://www.who.int/mental_health/management/depression/who_paper_depression_wfmh_2012.pdf, accessed on Dec. 19, 2021.

[3] Clarkin, J.F., Petrini, M., Diamond, D. (2019). Complex depression: The treatment of major depression and severe personality pathology. Journal of Clinical Psychology, 75(5): 824-833. https://doi.org/10.1002/jclp.22759

[4] Kharel, P., Sharma, K., Dhimal, S., Sharma, S. (2019). Early detection of depression and treatment response prediction using machine learning: A review. 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP).

[5] Wang, R., Chen, F., Chen, Z., Li, T., Harari, G., Tignor, S., Zhou, X., Ben-Zeey, D., Campbell, A.T. (2014). StudentLife: Assessing mental health, academic performance and behavioral trends of college students using smartphones. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and

Ubiquitous Computing, pp. 3-14. https://doi.org/10.1145/2632048.2632054
[6] Sobin, C., Sackeim, H.A. (1997). Psychomotor symptoms of depression. American Journal of Psychiatry, 154(1): 4-17.
[7] Sobin, C., Sackeim, H.A. (1997). Psychomotor symptoms of depression. American Journal of Psychiatry, 154(1): 4-17. https://doi.org/10.1176/ajp.154.1.4
[8] Low, L.S.A., Maddage, N.C., Lech, M., Sheeber, L., Allen, N. (2010). Influence of acoustic low-level descriptors in the detection of clinical depression in adolescents. In 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, USA, pp. 5154-5157. https://doi.org/10.1109/ICASSP.2010.5495018
[9] Jiang, H., Hu, B., Liu, Z., Wang, G., Zhang, L., Li, X., Kang, H. (2018). Detecting depression using an ensemble logistic regression model based on multiple speech features. Computational and Mathematical Methods in Medicine, Article ID: 6508319. https://doi.org/10.1155/2018/6508319
[10] Ramakrishnan, S., El Emary, I.M. (2013). Speech emotion recognition approaches in human computer interaction. Telecommunication Systems, 52(3): 1467-1478. https://doi.org/10.1007/s11235-011-9624-z
[11] Aloshban, N., Esposito, A., Vinciarelli, A. (2021). What you say or how you say it? Depression detection through joint modeling of linguistic and acoustic aspects of speech. Cognitive Computation, 1-14. https://doi.org/10.1007/s12559-020-09808-3
[12] Lee, S., Suh, S.W., Kim, T., Kim, K., Lee, K.H., Lee, J.R., Han, G., Hong, J.W., Han, J.W., Lee, K., Kim, K.W. (2021). Screening major depressive disorder using vocal acoustic features in the elderly by sex. Journal of Affective Disorders, 291: 15-23. https://doi.org/10.1016/j.jad.2021.04.098
[13] Samareh, A., Jin, Y., Wang, Z., Chang, X., Huang, S. (2018). Detect depression from communication: How computer vision, signal processing, and sentiment analysis join forces. IISE Transactions on Healthcare Systems Engineering, 8(3): 196-208. https://doi.org/10.1080/24725579.2018.1496494
[14] Low, L.S.A., Maddage, N.C., Lech, M., Sheeber, L.B., Allen, N.B. (2010). Detection of clinical depression in adolescents' speech during family interactions. IEEE Transactions on Biomedical Engineering, 58(3): 574-586. https://doi.org/10.1109/TBME.2010.2091640
[15] Toto, E., Tlachac, M.L., Stevens, F.L., Rundensteiner, E.A. (2020). Audio-based depression screening using sliding window sub-clip pooling. In 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, pp. 791-796. https://doi.org/10.1109/ICMLA51294.2020.00129
[16] Vázquez-Romero, A., Gallardo-Antolín, A. (2020). Automatic detection of depression in speech using ensemble convolutional neural networks. Entropy, 22(6): 688. https://doi.org/10.3390/e22060688
[17] Alghowinem, S.M., Gedeon, T., Goecke, R., Cohn, J., Parker, G. (2020). Interpretation of depression detection models via feature selection methods. IEEE Transactions on Affective Computing, 1-18. https://doi.org/10.1109/TAFFC.2020.3035535
[18] Rong, J., Li, G., Chen, Y.P.P. (2009). Acoustic feature selection for automatic emotion recognition from speech. Information Processing & Management, 45(3): 315-328. https://doi.org/10.1016/j.ipm.2008.09.003
[19] Haider, F., Pollak, S., Albert, P., Luz, S. (2021). Emotion recognition in low-resource settings: An evaluation of automatic feature selection methods. Computer Speech & Language, 65: 101119. https://doi.org/10.1016/j.csl.2020.101119
[20] Drotár, P., Gazda, M., Vokorokos, L. (2019). Ensemble feature selection using election methods and ranker clustering. Information Sciences, 480: 365-380. https://doi.org/10.1016/j.ins.2018.12.033
[21] Ostvar, N., Eftekhari Moghadam, A.M. (2020). HDEC: A heterogeneous dynamic ensemble classifier for binary datasets. Computational Intelligence and Neuroscience. https://doi.org/10.1155/2020/8826914
[22] He, L., Niu, M., Tiwari, P., Marttinen, P., Su, R., Jiang, J., Guo, C., Wang, H., Ding, S., Wang, Z., Pan, X., Dang, W. (2022). Deep learning for depression recognition with audiovisual cues: A review. Information Fusion, 80: 56-86. https://doi.org/10.1016/j.inffus.2021.10.012
[23] Rejaibi, E., Komaty, A., Meriaudeau, F., Agrebi, S., Othmani, A. (2022). MFCC-based recurrent neural network for automatic clinical depression recognition and assessment from speech. Biomedical Signal Processing and Control, 71: 103107. https://doi.org/10.1016/j.bspc.2021.103107
[24] Dai, Z., Zhou, H., Ba, Q., Zhou, Y., Wang, L., Li, G. (2021). Improving depression prediction using a novel feature selection algorithm coupled with context-aware analysis. Journal of Affective Disorders, 295: 1040-1048. https://doi.org/10.1016/j.jad.2021.09.001
[25] Zhang, P., Wu, M., Dinkel, H., Yu, K. (2021). DEPA: Self-supervised audio embedding for depression detection. In Proceedings of the 29th ACM International Conference on Multimedia, pp. 135-143. https://doi.org/10.1145/3474085.3479236
[26] Kroenke, K., Strine, T. W., Spitzer, R.L., Williams, J.B., Berry, J.T., Mokdad, A.H. (2009). The PHQ-8 as a measure of current depression in the general population. Journal of Affective Disorders, 114(1-3): 163-173. https://doi.org/10.1016/j.jad.2008.06.026
[27] Eyben, F., Weninger, F., Gross, F., Schuller, B. (2013). Recent developments in openSMILE, the munich open-source multimedia feature extractor. In Proceedings of the 21st ACM international conference on Multimedia, pp. 835-838. https://doi.org/10.1145/2502081.2502224
[28] Eyben, F., Scherer, K.R., Schuller, B.W., Sundberg, J., André, E., Busso, C., Devillers, L.Y., Epps, J., Laukka, P., Narayanan, S.S., Truong, K.P. (2015). The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing. IEEE Transactions on Affective Computing, 7(2): 190-202. https://doi.org/10.1109/TAFFC.2015.2457417
[29] Kohavi, R., John, G.H. (1997). Wrappers for feature subset selection. Artificial Intelligence, 97(1-2): 273-324. https://doi.org/10.1016/S0004-3702(97)00043-X
[30] Mukaka, M.M. (2012). Statistics Corner: A guide to appropriate use of Correlation coefficient in medical research. [Online]. Available: www.mmj.medcol.mw, accessed on 12 November 2021.
[31] Frank, E., Hall, M., Trigg, L., Holmes, G., Witten, I.H. (2004). Data mining in bioinformatics using Weka. Bioinformatics, 20(15): 2479-2481. https://doi.org/10.1093/bioinformatics/bth261

[32] Cai, H., Chen, Y., Han, J., Zhang, X., Hu, B. (2018). Study on feature selection methods for depression detection using three-electrode EEG data. Interdisciplinary Sciences: Computational Life Sciences, 10(3): 558-565. https://doi.org/10.1007/s12539-018-0292-5

[33] Breiman, L. (2001). Random forests. Machine Learning, 45(1): 5-32. https://doi.org/https://doi.org/10.1023/A:1010933404324

[34] Cacheda, F., Fernandez, D., Novoa, F.J., Carneiro, V. (2019). Early detection of depression: social network analysis and random forest techniques. Journal of Medical Internet Research, 21(6): e12554. https://doi.org/10.2196/12554

[35] Kursa, M.B., Rudnicki, W.R. (2010). Feature selection with the Boruta package. Journal of Statistical Software, 36: 1-13. https://doi.org/10.18637/jss.v036.i11

[36] Singh, K.P., Basant, N., Gupta, S. (2011). Support vector machines in water quality management. Analytica Chimica Acta, 703(2): 152-162. https://doi.org/10.1016/j.aca.2011.07.027

[37] Hu, C., Wang, J., Zheng, C., Xu, S., Zhang, H., Liang, Y., Bi, L., Fan, Z., Han, B., Xu, W. (2013). Raman spectra exploring breast tissues: Comparison of principal component analysis and support vector machine-recursive feature elimination. Medical Physics, 40(6Part1): 063501. https://doi.org/10.1118/1.4804054

[38] Yan, K., Zhang, D. (2015). Feature selection and analysis on correlated gas sensor data with recursive feature elimination. Sensors and Actuators B: Chemical, 212: 353-363. https://doi.org/10.1016/j.snb.2015.02.025

[39] Gu, Q., Li, Z., Han, J. (2012). Generalized fisher score for feature selection. arXiv preprint arXiv:1202.3725. https://arxiv.org/abs/1202.3725.

[40] Sim, J., Kim, S.Y., Lee, J. (2005). Prediction of protein solvent accessibility using fuzzy k-nearest neighbor method. Bioinformatics, 21(12): 2844-2849. https://doi.org/10.1093/bioinformatics/bti423

[41] Kittler, J., Hater, M., Duin, R.P.W. (1996). Combining classifiers. Proceedings of 13th International Conference on Pattern Recognition, Vienna, Austria, pp. 897-901. https://doi.org/10.1109/ICPR.1996.547205

[42] Brown, G., Wyatt, J., Harris, R., Yao, X. (2005). Diversity creation methods: a survey and categorisation. Information Fusion, 6(1): 5-20. https://doi.org/10.1016/j.inffus.2004.04.004

[43] Kuncheva, L.I., Whitaker, C.J. (2001). Ten measures of diversity in classifier ensembles: Limits for two classifiers. DERA/IEE Workshop Intelligent Sensor Processing, pp. 73-82. https://doi.org/10.1049/ic:20010105

[44] Kuncheva, L.I. (2002). A theoretical study on six classifier fusion strategies. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(2): 281-286. https://doi.org/10.1109/34.982906

[45] Dietterich, T.G. (2000). Ensemble methods in machine learning. In International Workshop on Multiple Classifier Systems, pp. 1-15. https://doi.org/10.1007/3-540-45014-9_1

[46] Ren, Y., Zhang, L., Suganthan, P.N. (2016). Ensemble classification and regression-recent developments, applications and future directions. IEEE Computational Intelligence Magazine, 11(1): 41-53. https://doi.org/10.1109/MCI.2015.2471235

[47] Polikar, R. (2006). Ensemble based systems in decision making. IEEE Circuits and Systems Magazine, 6(3): 21-45. https://doi.org/10.1109/MCAS.2006.1688199

[48] Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M. (2017). Ensemble learning for data stream analysis: A survey. Information Fusion, 37: 132-156. https://doi.org/10.1016/j.inffus.2017.02.004

[49] Kittler, J., Hatef, M., Duin, R.P., Matas, J. (1998). On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3): 226-239. https://doi.org/10.1109/34.667881

[50] Kuncheva, L.I., Whitaker, C.J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning, 51(2): 181-207. https://doi.org/10.1023/A:1022859003006

[51] Brown, G., Wyatt, J., Harris, R., Yao, X. (2005). Diversity creation methods: a survey and categorisation. Information fusion, 6(1): 5-20. https://doi.org/10.1016/j.inffus.2004.04.004

[52] Tang, E.K., Suganthan, P.N., Yao, X. (2006). An analysis of diversity measures. Machine learning, 65(1): 247-271. https://doi.org/10.1007/s10994-006-9449-2

[53] Richman, R., Wüthrich, M.V. (2020). Nagging predictors. Risks, 8(3): 1-26. https://doi.org/10.3390/risks8030083

[54] Bartlett, P., Freund, Y., Lee, W.S., Schapire, R.E. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. The Annals of Statistics, 26(5): 1651-1686. https://doi.org/10.1214/aos/1024691352

[55] Ho, T.K. (1995). Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, pp. 278-282. https://doi.org/10.1109/ICDAR.1995.598994

[56] Soares, R.G., Santana, A., Canuto, A.M., de Souto, M.C.P. (2006). Using accuracy and diversity to select classifiers to build ensembles. In The 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada, pp. 1310-1316. https://doi.org/10.1109/ijcnn.2006.246844

[57] Woods, K., Kegelmeyer, W.P., Bowyer, K. (1997). Combination of multiple classifiers using local accuracy estimates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4): 405-410. https://doi.org/10.1109/34.588027

[58] Sabourin, M., Mitiche, A., Thomas, D., Nagy, G. (1993). Classifier combination for hand-printed digit recognition. In Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93), Tsukuba, Japan, pp. 163-166. https://doi.org/10.1109/ICDAR.1993.395758

[59] Woloszynski, T., Kurzynski, M. (2011). A probabilistic model of classifier competence for dynamic ensemble selection. Pattern Recognition, 44(10-11): 2656-2668. https://doi.org/10.1016/j.patcog.2011.03.020

[60] Kuncheva, L.I., Rodriguez, J.J. (2007). Classifier ensembles with a random linear oracle. IEEE Transactions on Knowledge and Data Engineering, 19(4): 500-508. https://doi.org/10.1109/TKDE.2007.1016

[61] Ko, A.H., Sabourin, R., Britto Jr, A.S. (2008). From dynamic classifier selection to dynamic ensemble selection. Pattern Recognition, 41(5): 1718-1731. https://doi.org/10.1016/j.patcog.2007.10.015

[62] Britto Jr, A.S., Sabourin, R., Oliveira, L.E. (2014). Dynamic selection of classifiers—a comprehensive review. Pattern Recognition, 47(11): 3665-3680. https://doi.org/10.1016/j.patcog.2014.05.003
[63] Ko, A.H., Sabourin, R., Britto Jr, A.S. (2008). From dynamic classifier selection to dynamic ensemble selection. Pattern Recognition, 41(5): 1718-1731. https://doi.org/10.1016/j.patcog.2007.10.015
[64] Cruz, R.M., Sabourin, R., Cavalcanti, G.D., Ren, T.I. (2015). META-DES: A dynamic ensemble selection framework using meta-learning. Pattern Recognition, 48(5): 1925-1935. https://doi.org/10.1016/j.patcog.2014.12.003
[65] Cruz, R.M., Sabourin, R., Cavalcanti, G.D. (2017). META-DES. Oracle: Meta-learning and feature selection for dynamic ensemble selection. Information Fusion, 38: 84-103. https://doi.org/10.1016/j.inffus.2017.02.010
[66] Cruz, R.M., Sabourin, R., Cavalcanti, G.D. (2015). META-DES. H: A dynamic ensemble selection technique using meta-learning and a dynamic weighting approach. In 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, pp. 1-8. https://doi.org/10.1109/IJCNN.2015.7280594
[67] KP, M.N., Thiyagarajan, P. (2021). Alzheimer's classification using dynamic ensemble of classifiers selection algorithms: A performance analysis. Biomedical Signal Processing and Control, 68: 102729. https://doi.org/10.1016/j.bspc.2021.102729
[68] García, S., Zhang, Z.L., Altalhi, A., Alshomrani, S., Herrera, F. (2018). Dynamic ensemble selection for multi-class imbalanced datasets. Information Sciences, 445: 22-37. https://doi.org/10.1016/j.ins.2018.03.002
[69] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16: 321-357. https://doi.org/10.1613/jair.953
[70] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. The Journal of Machine Learning Research, 12: 2825-2830.
[71] Tazawa, Y., Liang, K.C., Yoshimura, M., Kitazawa, M., Kaise, Y., Takamiya, A., Kishi, A., Horigome, T., Mitsukura, Y., Mimura, M., Kishimoto, T. (2020). Evaluating depression with multimodal wristband-type wearable device: screening and assessing patient severity utilizing machine-learning. Heliyon, 6(2): e03274. https://doi.org/10.1016/j.heliyon.2020.e03274
[72] Hou, W.H., Wang, X.K., Zhang, H.Y., Wang, J.Q., Li, L. (2020). A novel dynamic ensemble selection classifier for an imbalanced data set: An application for credit risk assessment. Knowledge-Based Systems, 208: 106462. https://doi.org/10.1016/j.knosys.2020.106462
[73] Qi, B., Ramamurthy, J., Bennani, I., Trakadis, Y.J. (2021). Machine learning and bioinformatic analysis of brain and blood mRNA profiles in major depressive disorder: A case–control study. American Journal of Medical Genetics Part B: Neuropsychiatric Genetics, 186(2): 101-112. https://doi.org/10.1002/ajmg.b.32839
[74] Na, K.S., Cho, S.E., Geem, Z.W., Kim, Y.K. (2020). Predicting future onset of depression among community dwelling adults in the Republic of Korea using a machine learning algorithm. Neuroscience Letters, 721: 134804. https://doi.org/10.1016/j.neulet.2020.134804