Detecting Self-harm Content and Behavior in Tweets with SVM and Ensemble Classifiers: A Comparative Study

Divya Dewangan
Department of Computer Application,
Shri Shankaracharya Technical Campus, Bhilai, India,

Smita Selot
Department of Computer Application,
Shri Shankaracharya Technical Campus, Bhilai, India

Sreejit Panicker
Trainee Head,
Techment Technology, New STPI, Bhilai, India

ABSTRACT
Suicide is a mental state when the person loses the will to live, and it is a crucial issue nowadays. A person who is at suicidal risk needs early detection and medication. Researchers have exposed that many people relish posting their emotions and thoughts on social networking sites. In the past few years, Support Vector Machine (SVM) has been one of the most capable and vigorous classifiers in many fields of applications. Few areas where SVMs do not perform well have impelled the advancement of other applications, to enhance the strength of classifiers and parameters. Researchers from various disciplines have considered and explored the use of combination methodology. The idea of combination methodology is to build a prediction model by an ensemble of multiple methods. The main intention of this study is to develop and evaluate a model for suicidal content and behavior detection using the Support Vector Machine and Ensemble Classifiers such as Random Forest classifier, XGBoosting Classifier, and Stacking classifier. This paper summarized a brief introduction of SVM and Ensemble classifier. In this study, different Feature Extraction techniques, and their combinations have been used to train the models, and the accuracy and the performance of the models are analyzed. The findings highlight the substantial potential of the SVM and Ensemble classifier for accurately predicting suicidal content and behavior.

Keywords
Random Forest classifier, XGBoosting classifier, Latent Dirichlet Allocation, FastText Embeddings, Stacking classifier.

1. INTRODUCTION
World Health Organization (WHO) has revealed that on average, one person commits suicide every 40 seconds, and there are 20 attempts for each suicide, and approximately 800,000 suicides occur every year [1]. According to the survey of National Mental Health Survey of India, 2015-2016, there are 100,000 deaths by suicide every year in India and the death rate was 16.5 per, compared to the global average of 10.5 per 100,000 [2]. In India, the major resources by which suicide is attempted are burning himself, consuming poison, self-hanging, jumping off a bridge, and in front of railway tracks.

Many suicides occur out of passion in moments of anger. It causes the death of one person but it has a negative impact on the family and society. There must be a way to detect the thoughts and behaviors of the affected person quickly. If somehow identify that person, we can try to stop them from doing so with the help of family, friends, medications, counseling, etc. [3].

It adheres to a proper procedure in which suicidal ideation comes first and is followed by a suicidal attempt which then elongates into a completed suicide. Ideation and thinking do not lead to suicide but they create circumstances of prominent risk to individuals who may later attempt suicide [4]. American Psychological Association [APA] for Suicidal Behavior in Children and Adolescents identified different factors and warning signs of suicidal behavior to help probable individuals. They classify the risk factors into three main classes. These factors are related to Intrapersonal (mental disorder, serious loss, etc.), Social/situational (sense of isolation, family history, child abuse, etc.), and Cultural/environmental (access to lethal means, stigma, etc.) [6].

Social Networking platforms are nowadays very popular and openly accessible to everyone, where they interact, communicate, and post their views and thoughts. Researchers have found that people feel pleasure by posting their day-to-day life experiences on social platforms without thinking about social stigma [8]. Some Special keywords, features, or patterns are usually used by individuals having suicidal thoughts and need some advanced techniques to identify those features or patterns. In some studies, researchers have used Natural Language Techniques and Machine learning models to differentiate suicidal and non-suicidal content [13].

In this paper, machine learning models have been utilized to categorize the text into suicidal and non-suicidal content. Single and combination of feature extraction techniques have been used to train the models to differentiate the content into categories. In this paper, trained Support Vector Machine and Ensemble classifier are investigated on tweeter data. Support Vector machines, a Classification model, is a valuable tool in research due to their robust performance, flexibility, and interpretability, making them suitable for a wide range of applications across diverse domains [14]. Ensemble classifiers are a versatile and powerful tool in research and are used for various tasks such as sentiment analysis, text classification, and named entity recognition [17]. By combining the output of multiple classifiers trained on different features or using different algorithms, the overall performance of the models can be improved.

The major contribution to this proposed work is as follows:

- Evaluate the features of FastText, TFIDF, LDA, and LSA as a single and their combination with proposed models to achieve higher performance.
- Analyzing the SVM model and ensemble classifier models.

- Support Vector Machine (SVM), and Ensemble classifiers such as Random Forest (RF), Stacking classifier, and XGBoosting classifier are trained.
- Compare the accuracy and performance of SVM with ensemble classifiers based on a single feature and their combination with FastText.

The rest of the part is consolidated into sections. Section 2 represents the previous paper relevant to this topic, section 3 describes methodology, section 4 describes Experimental Analysis and section 5 describes conclusion.

2. LITERATURE REVIEW

There are various studies investigating the work related to SVM, Ensemble classifier, and suicidal behavior on social media, which provide new insight into machine learning techniques.

Burnap et al., (2017) worked on a multiclass classification to classify text related to suicidal data. They created a new human-annotated dataset to identify features and set a benchmark result for machine learning algorithms to differ in worrying languages such as suicidal thoughts and planning, and discourteous reference to suicide. They built baseline classifiers i.e. ensemble classifiers using the Rotation Forest algorithm and Maximum Probability voting classification and achieved an overall 0.78 F-measure [3].

Chadha and Kaushik (2019) examined social media data to analyze sentiments for the identification of suicidal thoughts and risk, and these various machine learning algorithms such as Bernoulli Naïve Bayes, Multinomial Naïve Bayes, Decision Tree, Logistic Regression, Support Vector Machine, and Ensemble Learning based algorithms such as Random Forest, AdaBoost, Voting Ensemble are experimented for solving identification and prediction problem of suicidal data [5].

Dewangan et al., (2023) have studied and analyzed suicide notes, online discussions, and social media posts to extract linguistic and content markers. The purpose of the study is to detect of signs of risk of self-harm/suicide in social media content. They used frequency-based featuring and prediction-based featuring to train different baseline machine learning models and it was found that SVM with FastText embedding achieved the best accuracy score [7].

Ghosal and Jain (2023) proposed and implemented a depression and suicidal risk content detection model. It classifies depression and suicidal content with two feature sets, FastText for contextual analysis and TFIDF vectorization for the relevance of terms, and machine learning model XGBoost classifier for classification [9].

Su et al., (2020) studied and examined a range of factors related to demographic, diagnostic, laboratory, and medication predictors derived from Electronic Health Records EHRs. Candidate predictors conducted statistical tests for each prediction window and then built a predictive model via a sequential forward feature selection procedure. This model predicted up to 62% of suicide-positive topics with 90% specificity [10].

Lekkas et al., (2021) worked on a previously published adolescent dataset on Instagram and mainly focused on linguistic and Social Networking data predicting acute suicidal ideation. They used predictors to capture language and activity used within social networking data and implemented ensemble learning methodologies such as XGBoosting, Logitboost, KNN, a three-layer feed-forward neural network, and aggregated and averaged random seed neural nets on the predictors for the prediction of acute suicidal ideation [11].

The work of Macalli et al., (2021) focused on developing a risk algorithm to examine the predictors for suicidal ideations and behaviors among students. They selected 70 potential predictors used with a Random Forest Classifier. They identified a group of predictors that can accurately predict students who have one-year suicidal ideation and behavior from the predictor assessment [13].

The work of Mahmud et al., (2023), is to build an efficient predictive model for suicidal behavior risk. For this, they conducted an online survey using a structured questionnaire to identify predictors such as socio-demographic, behavioral, and psychological features. The features are trained and tested using six machine learning models. Out of 6 learning models, SVM has achieved the best predictive algorithm [14].

Rabani et al., (2023) have proposed an Enhanced Feature Engineering Approach for Suicidal Risk Identification (EFASRI) to extract features and employed a multi-class machine learning classifier to identify suicidal risk levels in posts. They employed a Support Vector Machine, Random Forest, and XGBoosting classifier and it was observed that Extreme Gradient Boosting achieved the highest accuracy [17].

Song et al., (2022) employed ensemble learning models such as Extreme Gradient Boosting, CatBoost, and Light Gradient Boosting Method. They used clinical data and capture predictors to develop a suicide ideation prediction model that classify stroke patients into low-risk or high-risk [21].

Analysis of the Efficacy of SVM

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification and regression tasks. It works by finding the hyperplane that best separates data points into different classes in an n-dimensional space [24].

It has been successfully applied in several domains such as text classification, image recognition, and finance due to its versatility and effectiveness in handling both linear and non-linear classification problems [15].

Some mathematical explanation for the efficiency of SVM models.

a. **Optimization Objective**: The mathematical formulation of the SVM optimization objective involves minimizing the norm of the weight vector (W) subject to the constraint that all data points are correctly classified and lie in the correct side of the decision boundary (margin). By formulating the problem in this way, SVM aim to find a decision boundary that not only separates classes but also maximized the margin, leading to better generalization [4].

b. **Kernel Trick**: SVMs can handle non-linearly separable data by mapping input features into a higher-dimensional space using kernel functions. Mathematically, this can be understood as transforming the input feature space into a higher-dimensional feature space, where the data becomes linearly separable. The decision boundary in the higher-dimensional space is represented by a hyperplane [5]. Common kernel functions include

The linear kernel

$$\mathrm{K(x,y)} = x^T y,$$

Polynomial kernel

$$K(x, y) = (x^T y + c)^d$$

And Radial basis function (RBF) kernel

$$K(x, y) = exp(-\gamma\|x - y\|^2)$$

Where *x and y* are input feature vectors, c is a constant, d is the degree of the polynomial, and γ is a parameter controlling the smoothness of the decision boundary [4].

c. **Margin Maximization:** One of the key principles behind SVM is maximizing the margin between the decision boundary (hyperplane) and the nearest data points from each class. Mathematically, for a linear SVM, this can be represented as finding the optimal hyperplane that maximizes the distance between the hyperplane and the closest data points, known as support vectors. This margin is computed as the distance between the hyperplane and support vectors, which can be expressed mathematically using the concept of geometric margins and the norm of the weight vector (W) of the hyperplane.

$$\text{Margin} = \frac{2}{||W||}$$

Maximizing the margin leads to better generalization and robustness to overfitting, as the decision boundary is less likely to be influenced be noise or outliers [4].

d. **Regularization:** SVM consolidates a regularization parameter (*C*) that handles the trade-off between maximizing the margin and minimizing the classification error. Mathematically, this regularization term is added to the optimization objective and penalized large values of the weight vector (*W*) [24].

These mathematical aspects illustrate why SVM can be influential in certain scenarios, particularly when dealing with high-dimensional data, non-linear decision boundaries, and the need for margin maximization. But their popularity has somewhat waned due to a combination of factors including challenges with handling imbalanced data, their black box nature in complex scenarios, preprocessing requirements, and the availability of alternative algorithms that may offer better performance or ease of use in certain contexts. There are several factors that diminished the popularity in recent years:

a. **Interpretability:** SVMs, particularly when using complex non-linear kernels or high-dimensional feature spaces, can produce decision boundaries that are difficult to interpret or explain.

b. **Imbalance data handling:** SVMs can be less effective when dealing with highly imbalanced datasets where one class significantly outnumbers the other. While techniques like class weighting or using different cost functions can help address this issue to some extents, other algorithm like ensemble methods or specialized techniques for imbalanced data may offer better performance and ease of use in such scenarios [4].

c. **Kernel Selection and Tuning:** The choice of kernel function, and its associated parameters (such as *C* and *γ* for the RBF kernel) in SVM can significantly impact their performance. Selecting appropriate kernel and tuning its hyperparameters re quires careful experimentation and cross validation, which can be time-consuming and computationally expansive [5].

d. **Optimization challenges:** SVMs are based on solving a convex optimization problem with quadratic constraints. While techniques such as Sequential Minimal Optimization (SMO) [4] have been developed to efficiently solve the problem, it can still be challenging to find the global optimum, particularly for non-convex or highly-dimensional problems.

e. **Sensitivity to outliers:** SVMs are sensitive to outliers, as they aim to maximize the margin between classes. Outliers can significantly affect the position and orientation of the decision boundary, leading to suboptimal performance [4].

Analysis of the Efficacy of Ensemble Classifier

Ensemble methods like random forests [5] and gradient boosting machines [11] have gained popularity in recent years. Its popularity can be attributed to its ability to improve prediction accuracy, robustness to noise and outliers, interpretability, scalability, and proven performance across diverse applications and domains. Some key principles contribute to their effectiveness:

a. **Bias-Variance Tradeoff:** Ensemble classifiers mitigate the bias-variance tradeoff by combining multiple base classifiers. High-bias classifiers tend to have low variance but high bias, while high-variance classifiers exhibit the opposite behavior. By combining base classifiers with complementary biases and variance, ensemble methods can achieve better generalization performance than any single classifier alone [16].

b. **Law of Large Numbers:** The Law of large numbers states that as the number of independent observations increases, the average of these observations converges to the true expected value. Ensemble classifiers exploit this principle by aggregating predictions from multiple base classifiers, each trained on a random subset of the data or with different initialization parameters. As the number of base classifiers increases, the ensemble's predictive performance tends to approach the true underlying distribution of the data [16].

c. **Diversification:** Ensemble classifiers aim to maximize diversity among base classifiers to improve performance. Diversity is achieved through various mechanisms such as using different learning algorithms, varying the subset of training data, or introducing randomness during model training [18].

d. **Combining Weak Classifiers:** Ensemble methods often combine multiple weak classifiers to create a strong classifier. A weak classifier performs slightly better than random guessing [18]. By aggregating predictions from weak classifiers, ensemble classifiers can compensate for individual weaknesses and produce more accurate predictions than any single weak classifier alone.

e. **Error-Correction Mechanisms:** Ensemble classifiers employ error-correction mechanisms during prediction aggregation. For instance, in bagging, each base classifier is trained on a bootstrap sample of the training data, and predictions are combined by majority voting or averaging [16]. Bagging reduces the variance of the resulting ensemble by averaging out individual errors. In boosting, base classifiers are trained sequentially, with each subsequent classifier focusing more on instances that were misclassified by previous classifiers, thereby reducing bias and improving overall performance [18].

These mathematical principles provide a solid foundation for understanding why ensemble classifiers are effective and why they have gained popularity in machine learning. By leveraging the diversity, aggregation, and error-correction mechanisms of ensemble methods, practitioners can build robust and accurate predictive models across a wide range of domains and applications.

3. PROPOSED METHODOLOGY

In this study, a technical description of the approached were included and the framework illustrated in Fig 1, including data-preprocessing, feature extraction, machine learning classifiers, feature analysis and comparative experimental results.

3.1 Dataset Description and Exploration

The collected suicidal dataset is a suicidal tweet downloaded from the Kaggle website. This tweet contains a sample of 9119 tweets. It has two columns namely, tweet, and intention where tweet represents the twitter post and intention contains '0' and '1'. Here 0 represents non-suicidal tweets (5121) and 1 represents suicidal tweets (3998). The sample is split in the ratio of 70:15:15 for training, validation, and testing sets.

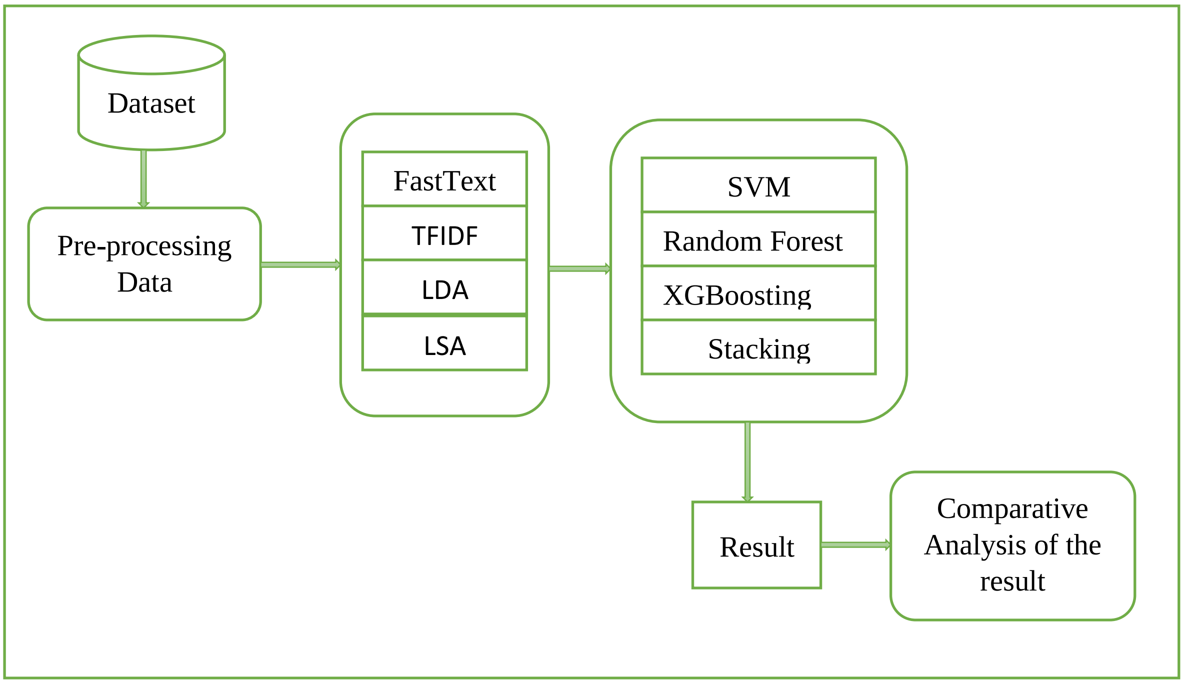


Fig. 1: Block Diagram of Different Classifiers

The dataset collected through the website is analyzed through Word Cloud illustrated in Fig 2. to get the frequent word used by individuals. Individuals with suicidal thoughts often use words such as "try", "die", "day", "never", "suicide", "help", and "live". Individuals with non-suicidal thoughts often use words such as "work", "much", "know", "god", "make", and "think".

3.2 Dataset Pre-processing and Feature Engineering

Before preprocessing, the dataset extracted from the website contains many irrelevant data and noise. It needs to filter and preprocess the data to make it reliable for further processing. After that, all the words are converted into lowercase form, to

(a) (b)

Fig 2: Word Cloud Exploration of Dataset. (a) Word Cloud of Suicidal Data, (b) Word Cloud of Non-Suicidal Data

filter the data. Data also contains URLs, punctuations, numbers, and stop words. The established methods like URLs, punctuation removal, and stop word removal, are applied to remove all the irrelevant data. To tokenize the data into tokens, RegexpTokenizer is used. Finally, all tokens are converted to their original form by applying WordNetLemmetizer.

Furthermore, the language of suicidal ideation lacks lexical and syntactic patterns. Therefore, feature engineering is required to analyze a set of features. After preprocessing the data, to detect users' linguistic usage in tweets, various feature extraction methods have been employed such as FastText Vectorization, TFIDF Vectorization [7], Latent Dirichlet Allocation (LDA) topics, and Latent Semantic Analysis (LSA). The features are described as follows:

3.2.1 FastText Vectorization:
FastText Vectorization extends the popular Word2Vec model by incorporating sub-word information. It can generate word embeddings (vectors) for words that can capture rich semantic and syntactic similarities between words including the ability to handle out-of-vocabulary words [7]. It considers each word as Bag-of-character n-grams and captures morphological and semantic information even for out-for-vocabulary words. It learns word embedding by training a neural network model and the model predicts the center word based on the context words or predicts context words based on the center word. After training, each word in the vocabulary is assigned a dense vector representation in a high-dimensional space. These vectors capture semantic and syntactic similarities between words. Words with similar meanings or contexts are represented by vectors that are closer together in this space [9].

3.2.2 Term Frequency Inverse Document Frequency:
TFIDF Vectorization is a technique used to convert text documents into numerical vectors [9] [17]. TF refers to the frequency of a term in a document and IDF refers to the rarity of a term across the entire corpus of documents. The TFIDF weighting of a term in a document is the product of its TF and IDF values. It combines information about the frequency of the term within the document (TD) and its rarity across the corpus to measure the importance of the term in the document relative to the corpus. A word represents a vector, and a dimension corresponds to a unique term in the vocabulary. The value of each dimension is the TFIDF weight of the corresponding term in the document [22].

3.2.3 Latent Dirichlet Allocation (LDA):
LDA topic modeling represents documents as vectors in a topic space where each dimension corresponds to a topic and the value at each dimension represents the documents' distribution across topics. This representation captures the underlying semantic structure of the documents [7] [9] [17]. By experimenting various topics, it is found that 10 was a reasonable value for accuracy and computational cost [12].

3.2.4 Latent Semantic Analysis:
Latent Semantic Analysis aims to capture the underlying semantic structure of a collection of documents by identifying hidden semantic relationships between words and documents. LSA vectorization represents documents as vectors in a reduced-dimensional semantic space, where each dimension captures a latent semantic concept [17].

It starts by constructing a TFIDF matrix. As the dataset grows, the TFIDF feature space can become huge, and highly sparse, especially when using techniques like n-grams. This problem leads to computational inefficiencies, the curse of dimensionality, and may hinder the performance of the Model. LSA addresses this issue by reducing dimensionality and capturing the latent semantic structure of the data through Singular Value Decomposition (SVD) [12].

3.3 Text Classification Techniques
To predict the presence of suicidal thinking in individuals, Machine Learning classifiers are employed to estimate the probability within the user. Support Vector Machine (SVM) and Ensemble classifiers such as Random Forest Classifier, Stacking classifier, and XGBoosting Classifier develop the proposed classification techniques.

3.3.1 Support Vector Machine (SVM) Classifier:
Support vector machine (SVM) is a supervised machine learning used for classification and regression tasks [3]. SVM works by finding an optimal hyperplane that separates data points of different classes with the largest margin. The hyperplane is a decision boundary that maximizes the distance between support vectors [7] [14] [17] (data points closest to the boundary) of each class. By maximizing the margin, SVM aims to achieve good generalization performance and robustness to new data [19].

3.3.2 Random Forest Classifier:
Random Forest is an ensemble learning method that constructs a multitude of decision trees during training [13]. Each tree is trained on a random subset of the training data, typically sampled with replacement (bootstrapping), and a random subset of features. The final prediction is made by averaging or voting the predictions of all individual trees [7] [17] [19].

3.3.3 Stacking Classifier:
A stacking classifier is a type of ensemble learning method in which a meta-classifier combines the predictions of multiple base classifiers to make the final prediction. The basic idea behind stacking is to use the strengths of multiple base classifiers to improve the overall accuracy and robustness of the classifier [23]. Combining the predictions of multiple classifiers allows the stacking classifier to identify patterns and relationships in the data that any single classifier may not capture.

The base classifiers used in the experiments are Decision Tree Classifier and Support Vector Machine and the meta-classifier is Logistic Regression. Firstly, the model splits the training data into a training set and a holdout set. Then train base classifiers with the training set and make predictions on the holdout set. Meta-classifier uses these predictions as input and by combining the prediction of the base classifier, the meta-classifier makes the final prediction of the test data.

3.3.4 XGBoosting Classifier:
XGBoost belongs to the family of gradient boosting algorithms [11] and is an ensemble technique where multiple weak learners are sequentially trained to correct the errors made by the preceding models. It builds an ensemble of decision tree, often referred to as "weak learners". Each decision tree is trained sequentially to minimize a differentiable loss function. It optimizes the ensemble of decision trees by minimizing a predefined objective function, which consists of two parts: a loss function and a regularization term [21]. The loss function quantifies the difference between predicted and actual values, while the regularization term penalizes the complexity of the model to prevent overfitting [17].

3.4 Tuning the hyperparameters of each classifier:

Tuning hyperparameters is crucial for optimizing the performance of machine learning models. Each classifier has its own set of hyperparameters that can be tuned to, and each classifier uses techniques like grid search, randomized search, or Bayesian optimization to search through the hyperparameter space efficiently. It finds the combination that results in the best performance on a validation set or through cross-validation. In the paper, hyperparameters have been defined for each model, and used GridSearchCV [7] object to search over hyperparameters. Thereby obtaining the best hyperparameters and the corresponding scores by accessing the 'best_estimator_' and 'best_score_' attributes. Given Table 1, shows the best hyperparameters obtained from each model.

Table 1. The best hyperparameters obtained from each model.

Name of Classifier	Defined Hyperparameters	best_estimator_		best_score_
SVM	param_grid = { 'alpha': [0.001,0.01,0.1,1], 'penalty': ['l1', 'l2','elasticnet'], 'max_iter': [200, 500], 'tol': [1e-3,1e-4], 'random_state': [42] }	FastText	{'alpha': 0.001, 'max_iter': 200, 'penalty': 'l1', 'random_state': 42, 'tol': 0.0001}	0.8260
		TFIDF	{'alpha': 0.001, 'max_iter': 200, 'penalty': 'elasticnet', 'random_state': 42, 'tol': 0.0001}	0.8699
		LDA	{'alpha': 0.001, 'max_iter': 500, 'penalty': 'l1', 'random_state': 42, 'tol': 0.001}	0.8692
		LSA	{'alpha': 0.001, 'max_iter': 200, 'penalty': 'elasticnet', 'random_state': 42, 'tol': 0.0001}	0.8728
		TFIDF+ FastText	{'alpha': 0.001, 'max_iter': 200, 'penalty': 'l2', 'random_state': 42, 'tol': 0.0001}	0.8852
		LDA+ FastText	{'alpha': 0.001, 'max_iter': 200, 'penalty': 'l2', 'random_state': 42, 'tol': 0.001}	0.8757
		LSA+ FastText	{'alpha': 0.001, 'max_iter': 200, 'penalty': 'elasticnet', 'random_state': 42, 'tol': 0.001}	0.8801
Random Forest Classifier	param_grid = { 'n_estimators': [100, 200], 'max_depth': [None], 'max_features': ['sqrt', 'log2'] }	FastText	(max_features='log2')	0.8969
		TFIDF	n_estimators=200	0.8845
		LDA	{'max_depth': None, 'max_features': 'log2', 'n_estimators': 200}	0.8567
		LSA	n_estimators=200	0.8604
		TFIDF+ FastText	{'max_depth': None, 'max_features': 'sqrt', 'n_estimators': 100}	0.8991
		LDA+ FastText	n_estimators=200	0.9137
		LSA+ FastText	max_features='log2'	0.9013
XGBoosting Classifier	param_grid = { 'n_estimators': [50, 100, 200], 'learning_rate': [0.01, 0.1, 0.5], 'max_depth': [3, 5, 10] }	FastText	{'learning_rate': 0.5, 'max_depth': 3, 'n_estimators': 100}	0.9006
		TFIDF	{'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 100}	0.8852
		LDA	{'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 50}	0.8728
		LSA	{'learning_rate': 0.5, 'max_depth': 3, 'n_estimators': 200}	0.8844
		TFIDF+ FastText	{'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 200}	0.9071
		LDA+ FastText	{'learning_rate': 0.5, 'max_depth': 10, 'n_estimators': 200}	0.9188

		LSA+ FastText	{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100}	0.9079

Stacking Classifier
# Base classifiers 	
base_classifiers = [
('dt', DecisionTreeClassifier(max_depth=3, random_state=42)), ('svm', SVC(kernel='linear', C=1.0, random_state=42))
]
# Meta-learner 	
meta_learner = LogisticRegression()
	param_grid = {
	'final_estimator__C': [0.1, 1.0, 10.0],
	
	'final_estimator__penalty': ['l2']
	}
		FastText	{'final_estimator__C': 0.1, 'final_estimator__penalty': 'l2'}	0.8348

		TFIDF	{'final_estimator__C': 1.0, 'final_estimator__penalty': 'l2'}	0.8720
		LDA	{'final_estimator__C': 10.0, 'final_estimator__penalty': 'l2'}	0.8669
		LSA	{'final_estimator__C': 1.0, 'final_estimator__penalty': 'l2'}	0.8757
		TFIDF+ FastText	{'final_estimator__C': 0.1, 'final_estimator__penalty': 'l2'}	0.8932
		LDA+ FastText	{'final_estimator__C': 1.0, 'final_estimator__penalty': 'l2'}	0.8998
		LSA+ FastText	{'final_estimator__C': 10.0, 'final_estimator__penalty': 'l2'}	0.8845

4. EXPERIMENTAL ANALYSIS

The study aims to do a comparative analysis of SVM algorithms with the top three ensemble classifiers i.e. the Random Forest classifier, Stacking classifier, and XGBoosting classifiers. The analysis of models is investigated by training the model with the suicidal dataset and comparing the performance with the test dataset. It considers 70% of the research dataset for training, 15% for validation, and the remaining 15% for testing purposes. This work has done by considering the 'no free lunch' theorem of machine learning, which states that no single algorithm can do well for all problems. As a result, the model is evaluated and validated through 10-fold cross-validation to reduce bias and variance. The Section describes evaluation metrics, model performance, results, and findings in the work.

4.1 Evaluation Metrics

Four metrics i.e. Precision (P), Recall (R), Accuracy (A), and F1-Measure are used to assess the performance of the model.

1. *Accuracy*: It measures the proportion of correctly classified instances among all instances in the dataset. It is a useful measure when the classes are balanced but it can be misleading if there's class imbalance.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

2 *Precision*: It measures the proportion of correctly predicted positive instances among all instances predicted as positive. It is important when the cost of false positives is high.

$$Precision = \frac{True\ Positive}{Sum\ of\ True\ Positive\ and\ False\ Positive}$$

3 *Recall*: It measures the proportion of correctly predicted positive instances among all actual positive instances. It is important when the cost of false negatives is high.

$$Recall = \frac{True\ Positive}{Sum\ of\ True\ Positive\ and\ False\ Negative}$$

4 *F1-Measure*: The F1-score is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall.

$$F1\text{-}score = \frac{2*(\boldsymbol{Precision*recall})}{(\boldsymbol{precision+recall})}$$

F1-score ranges from 0 to 1, where the higher value indicates better model performance.

4.2 Model Performance

To start the execution of the different classification techniques, the different dimension feature space and their combination extracted from the dataset have been used. The base features such as FastText, TFIDF [20], Latent Dirichlet Allocation (LDA) [12], and Latent Semantic Analysis (LSA) [12] and their multiple combination with FastText are built on a Twitter training dataset. Combining the features with FastText aims to investigate which feature combination gives the best performance accuracy of models.

Python has been used to generate features and implement models. The dataset is presented as training, validation, and testing set in the ratio 70:15:15 but the technique of partitioning the dataset suffers from inherent problems of bias and variance. For this, a 10 fold cross-validation technique are applied that assures the model predicts the correct patterns in the data. Performance evaluation of single features and combined features with the classifiers are shown in Table 2 which shows the Performance Analysis of SVM with an Ensemble classifier. In the table, accuracy and F1-score values are equivalent because the dataset is balanced.

Table 2. The Performance Analysis of SVM with an Ensemble classifier

Name of Classifier	Feature Extraction Techniques	Accuracy	Precision	Recall	F1-Score
SVM	FastText embedding	0.8355	0.8144	0.8362	0.8355
	TFIDF	0.8815	0.9421	0.7937	0.8816

	LDA	0.7346	0.8908	0.4882	0.7346
	LSA	0.8662	0.9414	0.7591	0.8662
	TFIDF + FastText	0.8896	0.9498	0.8047	0.8896
	LDA + FastText	0.5972	1.0	0.1323	0.5972
	LSA + FastText	0.8852	0.8746	0.8787	0.8852
Random Forest Classifier	FastText embedding	0.5358	0.72	0.041	0.5358
	TFIDF	0.8882	0.9288	0.822	0.8882
	LDA	0.8158	0.9101	0.6693	0.8158
	LSA	0.8582	0.8744	0.811	0.8582
	TFIDF + FastText	0.5534	1.0	0.0378	0.5534
	LDA + FastText	0.7661	0.707	0.8472	0.7661
	LSA + FastText	0.5373	0.75	0.0047	0.5373
XGBoosting Classifier	FastText embedding	0.5373	0.75	0.0047	0.5373
	TFIDF	0.8999	0.9415	0.8362	0.8999
	LDA	0.8202	0.9598	0.6394	0.8202
	LSA	0.8706	0.8895	0.8236	0.8706
	TFIDF + FastText	0.7281	1.0	0.4142	0.7281
	LDA + FastText	0.8341	0.9789	0.6567	0.8341
	LSA + FastText	0.5409	1.0	0.011	0.5409
Stacking Classifier	FastText embedding	0.6732	0.9087	0.3291	0.6732
	TFIDF	0.8925	0.9281	0.8331	0.8925
	LDA	0.8129	0.9417	0.6362	0.8129
	LSA	0.8794	0.9211	0.8094	0.8794
	TFIDF + FastText	0.8662	0.9768	0.7291	0.8662
	LDA + FastText	0.8794	0.8983	0.8346	0.8794
	LSA + FastText	0.8545	0.9739	0.7055	0.8545

5. RESULTS AND DISCUSSION

Table 2 shows the accuracy result of SVM models with Ensemble classifiers. Each model contains accuracy, precision, recall, and F1-score values. By evaluating the performance of classifiers with single features (FastText, TFIDF, LDA, and LSA), it can be observed that the best accuracy achieved by TFIDF with XGBoosting classifier with 89.99% accuracy followed by the Stacking classifier with 89.25% accuracy, followed by SVM with 88.15% accuracy. Among the four machine learning algorithms, the Stacking classifier outperformed the other three algorithms. By evaluating the performance of classifiers with combined features, it can be observed that the best accuracy achieved classifier is SVM with TFIDF + FastText feature with 88.96% accuracy, followed by SVM with LSA + FastastText feature with 88.52% accuracy followed by stacking with LDA + FastText feature with 87.94% accuracy.

But keeping in mind the purpose of the experiment, we need to examine the performance of each classifier with each feature and how well the classifier performs for each feature. To examine each classifier, Fig 3 illustrates accuracy Chart of SVM and Ensemble Classifiers for different feature extraction techniques. It can be seen that the performance of the stacking classifier is better than SVM [19] and others. It performed well with single as well as combined features. The main reason why Stacking performs better than others is that it combines predictions from diverse base models. The use of SVM and Decision Tree as base classifiers and logistic regression as a meta-learner introduces diversity into the underlying model. This combination of predictions can effectively leverage the strengths of each base model, capture different aspects of the data, and potentially lead to better generalization performance compared to a single SVM and others. Furthermore, aggregating the predictions from multiple base models helps reduce variance and improve robustness. XGBoosting classifier with single features also performs much better than Stacking and SVM.

Random Forest classifier [20], XGBoosting classifier performed very poorly with FastText features. FastText vectorizer generates dense word embeddings that represent the semantic words in a text corpus and have high dimensionality. However, Random Forest classifiers typically work better with sparse, binary, or count-based features and XGBoosting classifier has the capability of capturing complex relationships in the data. Thus the dense nature of FastText leads to decreased predictive performance and increased computational cost.

The comparative analysis of the prediction accuracy of all the classifiers can also be shown as the AUC-ROC curve. The plotted AUC-ROC curve helps in investigating the execution of each classifier with single and combined features individually [7] as shown in Fig 4. It is a mechanism that helps to visually compare and summarize the performance of a classifier across all possible classification thresholds. It plots the x-axis and the y-axis on the false positive rate against the true positive rate respectively.

The trade-off refers to the balance between the true positive rate and the false positive rate at different classification thresholds. By varying the probability thresholds for classifiers, the trade-off changes [9]. A higher AUC-ROC score indicates better performance; when comparing the multiple models, the one with the highest score will be preferred. From Fig 4, by observing each plot, it can be concluded that the stacking classifier has the highest AUC score and ROC plot in each graph. Stacking classifier showed

the highest AUC score of 0.96 for the TFIDF + FastText feature.

6. Conclusion

Comparative analysis between SVM and ensemble classifiers like Random Forest, XGBoosting, and Stacking classifiers has provided a better understanding of each classifier's strengths and limitations. It also provides valuable insights into their performance characteristics and suitability for specific tasks. The SVM model tends to provide more interpretable models, as a subset of training instances represents the decision boundary. In contrast, Ensemble classifiers are often considered “Black Box” models, making interpreting the learned decision rules challenging. SVM has fixed model complexity determined by the kernel function and regularization parameters, whereas the complexity of ensemble classifiers can be adjusted by modifying hyperparameters such as the number of trees, depth, or learning rate. This flexibility allows ensemble classifiers to adapt to the complexity of the dataset. The aggregation of multiple base models in ensembles helps mitigate the impact of individual noisy or outlier data points. The findings of this research work offer valuable insights for SVM, and ensemble classifiers based on experimental work. A single SVM model has many boundaries but combining SVM with other models can advantages of it, become more robust, and increase performance. Some work can be considered as future directions listed under:

- The experiment can be conducted on rigorous benchmarking experiments to compare the performance of classifiers across multiple datasets and evaluation metrics.
- Experiment with advanced ensemble methods based on neural networks.
- Explore different feature engineering techniques to enhance the performance of classifiers, including dimensionality reduction, feature selection methods, and feature transformation techniques.

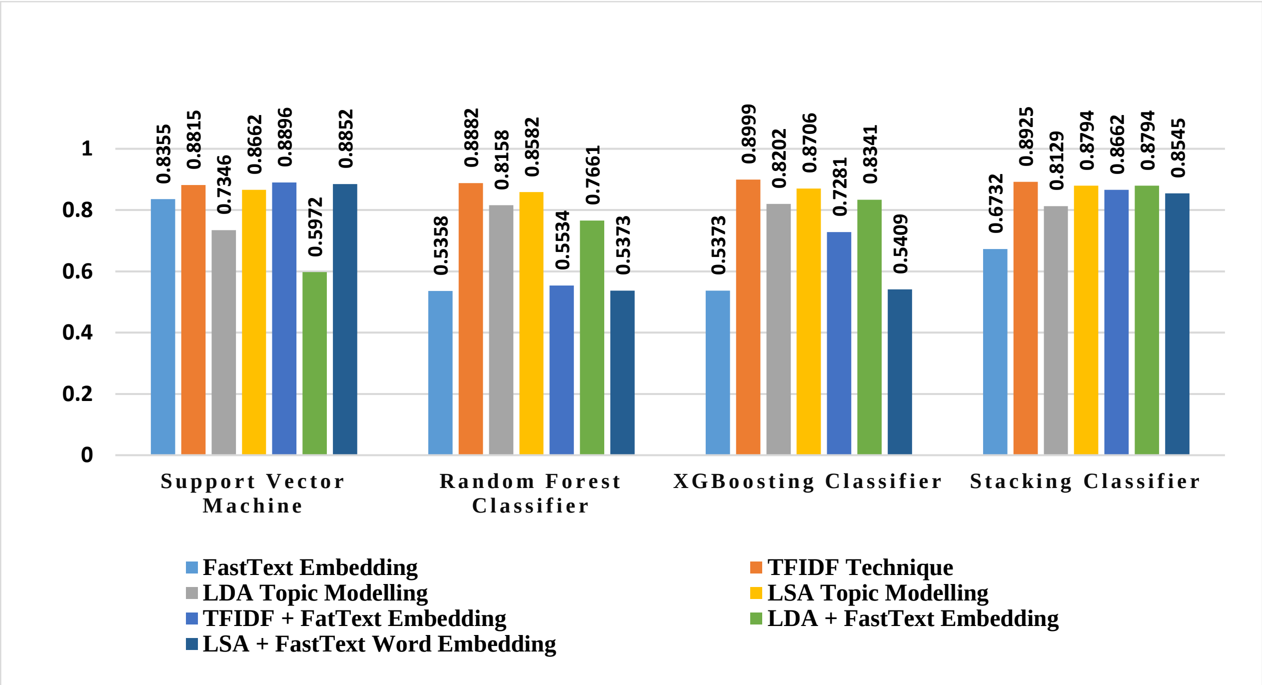
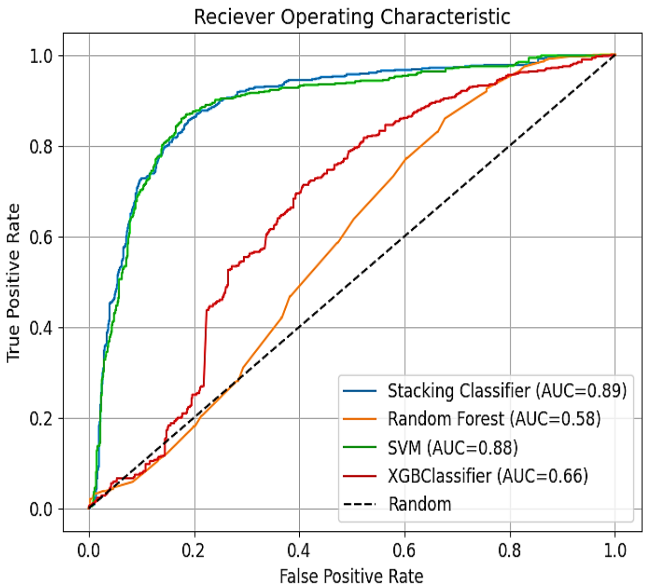
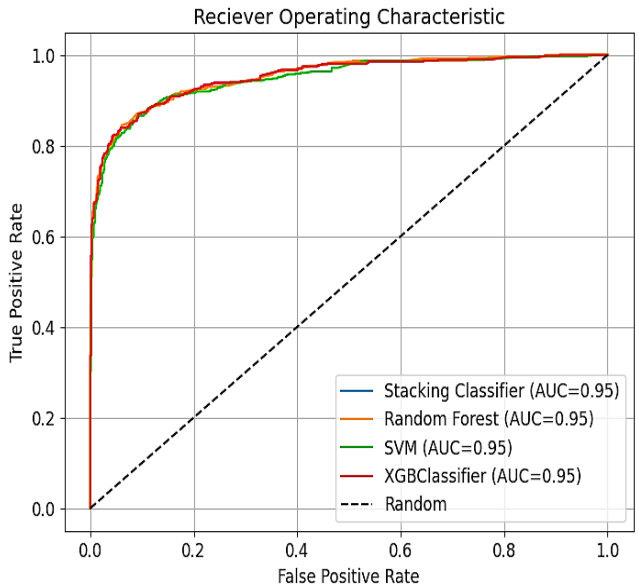


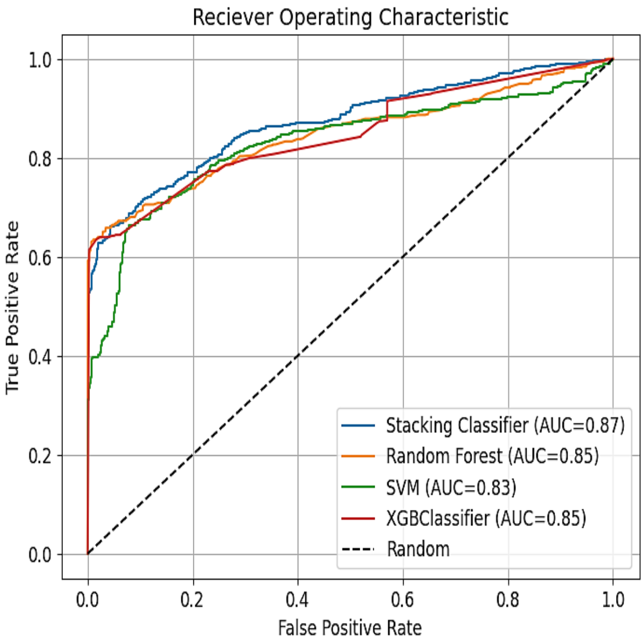
Fig 3: Accuracy Chart (in percentage) of SVM and Ensemble Classifiers for different feature extraction techniques



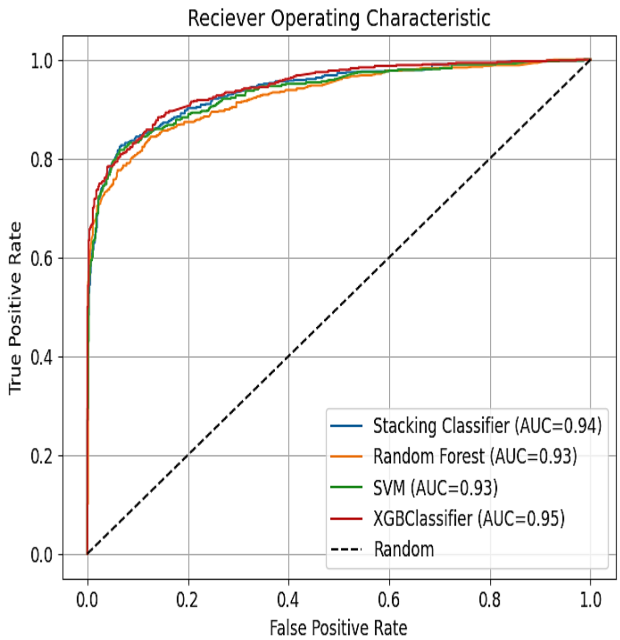
a) AUC-ROC curve of the classifiers with FastText Word embedding technique



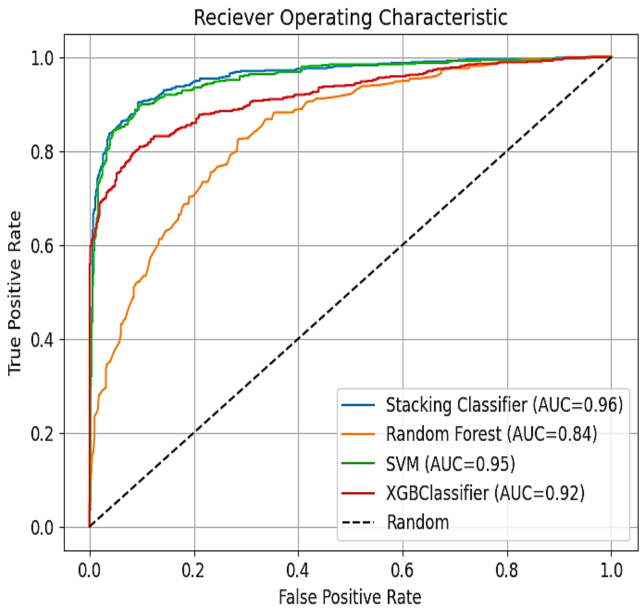
b) AUC-ROC curve of the classifiers with TFIDF technique



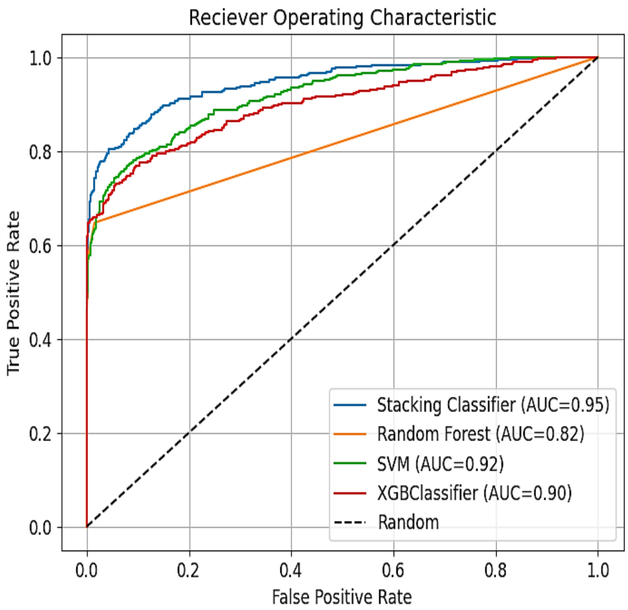
c) AUC-ROC curve of the classifiers with LDA technique



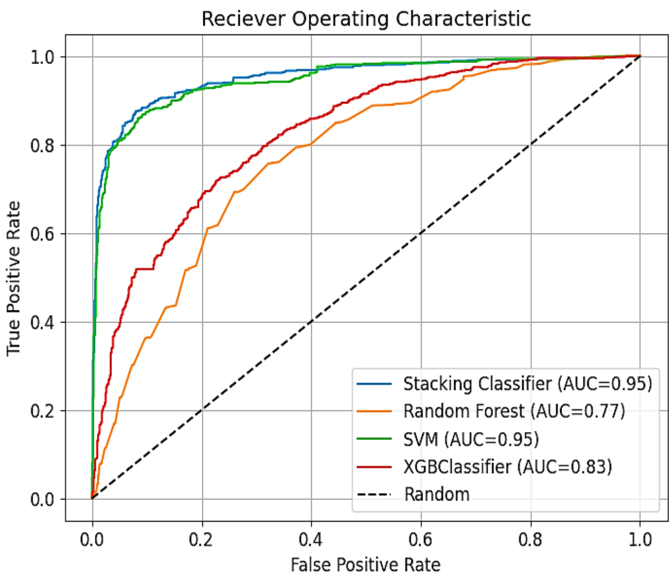
d) AUC-ROC curve of the classifiers with LSA technique



e) AUC-ROC curve of the classifiers with FastText + TFIDF technique



f) AUC-ROC curve of the classifiers with FastText + LDA technique



g) AUC-ROC curve of the classifiers with FastText + LSA technique

Fig 4: Plotting and comparing the AUC-ROC curve for each classifiers with different feature extraction techniques

7. REFERENCES

[1] World Health Organization, Suicide; 2023 https://www.who.int/news-room/fact-sheets/detail/suicide. Accessed on 11 December 2023

[2] National Mental Health Survey of India, Prevalence, Patterns and Outcomes, Supported by Ministry of Health and Family Welfare, Government of India, and Implemented by National institute of Mental Health and Neurosciences (NIMHANS) Bengaluru: In Collaboration with Partner Institutions; 2015-2016: https://aiimsbhopal.edu.in/AIIMSFiles/officialDownload/nmhs.pdf..

[3] Burnap, P., Colombo, G., Amery, R., Hodorog, A. and Scourfield, J. 2017. Multi-class machine classification of suicide-related communication on twitter. Online Social Networks and Media, 2: 32-44.

[4] Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L. and Lopez, A. 2019. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing.

[5] Chadha, A. and Kaushik, B. (2019) A Survey on Prediction of Suicidal Ideation Using Machine and Ensemble Learning. The British Computer Society. Computational Intelligence, Machine learning and Data Analytics, the Computer Journal.

[6] Kaslow, N. J. 2014. Suicidal Behavior in Children and Adolescents-[APA]. Available at: https://www.apa.org/about/governance/president/suicidal-behavior-adolescents.pdf. Accessed: 21December2023.

[7] Dewangan, D., Selot, S. and Panicker, S. 2023. The accuracy analysis of Different Machine Learning Classifiers for Detecting Suicidal Ideation and Content. Asian Journal of Electrical Sciences, 12(1):46-56.

[8] Littilefield, A. K., Cooke, J. T., Bagge, C. L., Glenn, C. R., Kleiman, E. M., Jacobucci, R., Millner, A. J. and Steinley, D. 2021. Machine learning to classify Suicidal Thoughts and Behaviors: Implementation within the Common Data Elements Used by the Military Suicide Research Consortium. Clinical Psychological Science, 9(3):467-481.

[9] Ghosal, S. and Jain, A. 2023. Depression and Suicide Risk Detection on Social Media using FastText Embedding and XGBoost classifier. In Proceedings of International Conference on Machine Learning and Data Engineering, Procedia Computer Science, 2023, 218:1631-1639.

[10] Su, C., Aseltine, R., Doshi, R., Chen, K., Rogers, S. C. and Wang, F. 2020. Machine learning for suicide risk prediction in children and adolescents with electronic health records. Translation Psychiatry, 10(413).

[11] Lekkas, D., Klein, R. J. and Jacobson, N. C. 2021. Predicting acute suicidal ideation on instagram using ensemble machine learning models. Internet Interventions, 25(100424).

[12] Gandhi, A. 2021. Topic Modeling with LSA, PLSA, LDA & lda2Vec: Nanonets; 2021 https://nanonets.com/blog/topic-modeling-with-lsa-plsa-lda-lda2vec/#lda-in-deep-learning-lda2vec.

[13] Macalli, M., Navarro, M., Orri, M., Tournier, M., Thiébaut, R., Côté, S. M. and Tzourio, C. A. 2021. machine learning approach for predicting suicidal thought and behaviors among college students. Scientific Reports, 11(11363).

[14] Mahmud, S., Mohsin, M., Muyeed, A., Nazneen, S., Sayed, A., Murshed, N. and Tonmon, T. T. 2023. Islam A. Machine learning approaches for predicting suicidal behaviors among university students in Bangladesh during the COVID-19 pandemic-A cross-sectional study. Medicine, 102(28).

[15] Alexandropoulos, S. A. N., Aridas, C. K., Kotsiantis, S. B. and Vrahatis, M. N. 2019. Stacking Strong Ensembles of Classifier. In Proceedings of the International federation for Information Processing, 559:545-556.

[16] Ngo-Yel, T. and Dutt, A. 2009. A Study on Efficacy of Ensemble Methods for Classification Learning. In Proceedings of International Conference on Information Systems, Procedia ICIS 2009 Proceedings.

[17] Rabani, S. T., Khanday, A. M. U. D., Khan, Q. R., Hajam, U. A., Imran, A. S. and Kastrati, Z. 2023. Detecting suicidality on social meadia: Machine learning at rescue. Informatics Journal, 24:291-302.

[18] Rokach, L. 2010. Ensemble-based Classifiers. Artif Intell Rev, 33:1-39. DOI: 10.1007/s10462-009-9124-7.

[19] Selot, S. and Panicker, S. 2021. Comparative performance of Random Forest and Support vector Machine on Sentiment Analysis of Reviews of Indian Tourism. IT in Industry, 9(2).

[20] Salehi, M., Ghahari, S., Hosseinzadeh, M. and Ghalichi, L. 2023. Domestic violence risk prediction in Iran using a machine learning approach by analyzing Persian Textual content in social media. Heliyon, *Vol.* 9.

[21] Song, S., Hong, H. T., Lee, C. and Lee, S. B. 2022. A Machine learning approach for predicting suicidal ideation in post stroke patients. Scientific Reports, 12(15906).

[22] Chatterjee, M., Kumar, P., Samanta, P. and Sarkar, D. 2022. Suicide ideation detection from online social media: A multi-modal feature based technique, International Journal of Information Management Data Insights, 2(2).

[23] Al-Zahrani, F. A., Abdulrazak, L. F., Ali, M. M., Islam, M. N. and Ahmad, K. 2023. StackDPP: Stacking-Based Explainable Classifier for Depression prediction and Finding the risk Factors among Clinicians. Egyptian, Bioengineering, 10:858.

[24] Wu, Q. and Zhou, D. X. 2006. Analysis of support vector machine classification. Journal of Computational Analysis and Applications.