IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 August 2024, accepted 2 September 2024, date of publication 5 September 2024,
date of current version 16 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3454796

RESEARCH ARTICLE

A Suicidal Ideation Detection Framework on Social Media Using Machine Learning and Genetic Algorithms

ABDALLAH BASYOUNI[1], HATEM ABDULKADER[1], WAIL S. ELKILANI[2], ABDULLAH ALHARBI[3], YULONG XIAO[4], AND ASMAA H. ALI[1]

[1]Information Systems Department, Faculty of Computers and Information, Menoufia University, Shebeen El-Kom 32511, Egypt
[2]College of Applied Computer Science, King Saud University, Riyadh 11421, Saudi Arabia
[3]Department of Computer Science, Community College, King Saud University, Riyadh 11437, Saudi Arabia.
[4]School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Corresponding author: Abdallah Basyouni (abdallah.basyounia@ci.menofia.edu.eg)

This work was supported by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project RSP2024R444.

ABSTRACT Suicide is a serious problem that affects modern societies all over the world, and its prevention is critical. Suicidal ideas are influenced by a variety of risk factors, such as depression, anxiety, hopelessness, and social isolation. Early detection of these risk factors can significantly reduce or prevent suicide attempts. Online platforms, particularly social media, have become a popular outlet for young people to express suicidal ideation. However, detecting and responding to such ideation effectively poses significant challenges in natural language processing (NLP) and psychology. To improve the efficiency of detecting suicidal ideation, this study proposes a novel framework that overcomes prior research limitations, such as feature redundancy and limited relevance to the target class. The framework addresses these issues by extracting meaningful and context-related features from posts that capture contextual and semantic aspects. Furthermore, a genetic algorithm is used to select the most important and relevant features that are strongly associated with the target class while excluding those that are redundant or irrelevant. To evaluate the effectiveness of the proposed framework, the following machine learning classifiers were used to determine whether a post indicates suicidal ideation or not: Random Forest (RF), Naive Bayes (NB), gradient boost classification tree (GBDT), and XGBoost. The proposed framework's results outperformed previous research, demonstrating the framework's high efficiency. The best performance in our study was achieved using the Random Forest (RF) classifier, which was applied to the features selected by the Genetic Algorithm (GA) from the linguistic feature set. When evaluated using 5-fold cross-validation, this approach yielded an impressive accuracy rate of 98.92% and an F1-score of 98.92%. These outstanding performance metrics demonstrate the framework's effectiveness in accurately detecting suicidal ideation.

INDEX TERMS Mental health, suicide detection, Reddit social media, feature selection, genetic algorithm.

I. INTRODUCTION

Mental health is a worldwide concern. According to the World Bank, at least 10% of the world's population suffers from mental health problems.[1] Suicide attempts are significantly increased by mental health issues. Suicide has been recognized as a significant public health issue on a global scale. According to the US National Alliance on Mental Illness, 46% of suicide victims had mental health issues[2] Suicide ranks as the fourth leading cause of death among individuals aged 15 to 29, resulting in approximately

The associate editor coordinating the review of this manuscript and approving it for publication was N. Ramesh Babu.

[1]https://www.worldbank.org/en/topic/mental-healt

[2]NAMI report on Risk Of Suicide, available at https://www.nami.org/Learn-More/Mental-Health-Conditions/Related-Conditions/ Suicide

800,000 deaths worldwide annually, a suicide occurs every 40 seconds, according to the World Health Organization (WHO) in 2019 [1]. Previous research on suicide analysis has predominantly focused on the psychological and clinical aspects of the phenomenon [2]. Multiple studies have recently focused on natural language processing approaches and machine learning [3], [4], [5]. These studies, however, have limitations. (1) Although gathering data on patients is very expensive from a psychological standpoint, some online data may aid in understanding suicidal thoughts. (2) Simple feature sets are insufficient to detect suicidal ideation, necessitating the development of new detection methods. Language preferences could be used to predict mental health status, which can help detect suicidal ideation early [6], [7].

Social media has become deeply ingrained in the lives of people. People spend a significant amount of time on social media. Social media or online communities such as Twitter, Weibo, Reddit, and Facebook are rapidly growing [4]. With this growth comes an increased desire among people to form online communities and connect to share their ideas. Because they prefer to remain anonymous, psychiatric patients do not prefer to see a psychiatrist [8]. With the option for the user to conceal his identity on social media. It allowed him to express his suicidal thoughts without fear of being discovered [9]. Research indicates that individuals tend to express their thoughts about suicidal ideation more readily online rather than in face-to-face interactions [8]. The feeling of restraint and the uncertain nature of social media platforms contribute to the accumulation of vast quantities of social data that hold valuable insights into people's interests, moods, and behavior [10]. Although social media is one of the richest mines of textual content, gathering this data is difficult due to user privacy. There have been several cases where suicide victims have re-ported their final thoughts before death on social media [11]. Due to the absence of laboratory experiments for predicting suicidal ideation, there is a belief that social media could offer an opportunity to analyze individuals' behavior by identifying early risk factors and warning signs [12]. This analysis could potentially aid in preventing deaths by taking timely intervention measures. Depression, anxiety, hopelessness, and stress are the primary causes of suicidal ideation, which gradually progresses to suicidal behavior [13]. If these risk factors are sufficiently analyzed on social media, it can help to prevent potential suicidal victims.

Text classification, a widely used machine learning application, entails automatically categorizing textual data into predefined classes [14]. Given the exponential growth of digital text documents, text classification finds applications in various domains [15]. To employ machine learning algorithms on text documents, it is essential to first perform text preprocessing using NLP techniques like tokenization, stemming, and stop-word removal [16]. The bag-of-words model is the predominant approach employed in text classification for representing data [17]. In this model, the focus is primarily on the sequence of terms within a document, disregarding the arrangement, structure, context, and meaning of the text. The resulting dataset, denoted as $D = \{d^t, C^t\}_{t=1}^{N}$, consists of N documents and M terms. Each document, represented as $d^t \epsilon R^M$ is associated with a category $C^t \epsilon \{\pm 1\}$. The set $T = \{T_1, T_2, \ldots, T_M\}$ represents the M distinct terms. However, it is important to note that this model entirely overlooks the contextual and semantic aspects of the text. While the bag-of-words model focuses on the presence or absence of individual words in a text, it disregards the context provided by surrounding words. To address this limitation, the proposed framework utilizes the n-gram model, which captures contiguous sequences of words. By employing the n-gram model, it becomes possible to extract meaningful vector-based features from text data. These features can be represented by a term's weight, such as its term frequency (tf) or term frequency-inverse document frequency (tf-idf) [18]. The feature space of a text classifier comprises a vast set of words, denoted by M, which poses a computational challenge when training the classifier [19]. High dimensionality in a dataset can lead to a decrease in classifier accuracy. This is primarily due to the presence of redundant and irrelevant features [20]. Irrelevant or redundant features do not provide any additional insights beyond what relevant features already convey about the class. Including these redundant and irrelevant features can mislead the learning algorithm [21]. To mitigate this issue, feature selection techniques can be employed [22]. Feature selection involves mapping the original feature space with M features to a new space by selecting the most important m dimensions from the original space while preserving the original meaning of the features [23].

Reddit is an internet platform dedicated to hosting discussions where individuals can express their opinions, emotions, and personal issues, including matters related to their family. Within Reddit, there are specific sections called subreddits that focus on particular subjects. One such subreddit is called "SuicideWatch (SW)," which serves as a space for individuals to openly talk about and exchange their thoughts and intentions regarding suicide [24]. The identification of suicidal ideation involves the analysis of written text to determine if a person is experiencing such thoughts. Social media platforms can be utilized to detect patterns in users' content that may indicate suicidal ideation. In our research, we collected textual posts from the Reddit subreddit "SuicideWatch." To comprehend suicidal ideation through the lens of textual data analysis, it is necessary to examine the posts, language patterns, and engage in topic modeling.

This paper aims to present a unique framework for detecting suicidal ideation on Reddit. The framework consists of a series of sequential stages, including data collection, data preprocessing, and feature engineering, which encompasses two sub-stages: feature extraction and feature selection,

followed by the classification phase. The framework offers several benefits compared to alternative methods by addressing limitations in these studies by overcoming the focus solely on word-level analysis. This is achieved by extracting contextually relevant features, including TF-IDF, topics, and n-gram features, in the feature extraction phase. Additionally, the problem of using all extracted features without considering their individual importance is resolved by employing a genetic algorithm for feature selection, which selects relevant and significant features. The primary contributions of this paper are as follows:

- Identifying the limitations of previous research on detecting suicidal ideation on social media and proposing a novel framework to overcome these challenges.
- Developing a framework that extracts meaningful and context-related features from social media posts, addressing the issues of feature redundancy and limited relevance to the target class.
- Implementing a genetic algorithm for feature selection, ensuring that only the most important and relevant features are included in the analysis.
- Demonstrating the effectiveness of the proposed framework through evaluation on relevant datasets and comparison with state of the art techniques.

The paper is organized as follows: Section II discusses the related work of suicide ideation detection, Section III presents the materials and methods used in the proposed framework. Section IV showcases the results obtained from applying the proposed framework to the dataset. Finally, Section V provides the conclusion of the paper.

II. RELATED WORK

In Literature, the increasing rates of suicide in recent years have sparked significant interest among researchers in the identification of suicidal individuals. Suicide is a complex issue influenced by various risk factors, necessitating the utilization of diverse research methods and techniques. Generally, the related works can be classified into several categories.

A. STANDARD METHODS

Standard methods which includes clinical methods and classical methods. Clinical methods may examine resting-state heart rate [25] and event-related instigators [26] Classical methods also include using questionnaires to assess the potential risk of suicide and applying clinician patient interactions [27].

B. MACHINE LEARNING

Machine learning methods also play a role in this field, utilizing key features such as knowledge-based features, N-grams, syntactic features, class-specific features, and contextual features [28].

Recently, there has been a spike in interest in the use of artificial intelligence approaches to detect suicidal ideation from a variety of sources, including social media [29]. Liu et al. [30] conducted a study where they utilized basic statistical characteristics (BSC), word embedding clustering (WEC), and risk factors for suicide (RFC) as extracted features. Additionally, they proposed an ensemble strategy that combines these features for the purpose of detecting suicidal thoughts on the social media platform Weibo. Haque et al. [31] conducted a comparative study where they analyzed various machine learning and deep learning models in order to detect suicidal thoughts extracted from the social media platform Twitter. Chatterjee et al. [32] present a novel approach to detecting suicidal ideation in social media data. The study utilizes various features such as linguistic, statistical, sentiment, topic, emoticons, and temporal features to detect suicide ideation in Twitter data. The study achieved an accuracy of 87% using the Logistic Regression classifier. Rabani et al. [33] present a machine-learning approach along with ensemble methods, aiming to detect suicidal ideations specifically on the Twitter platform. Rajesh Kumar et al. [34]. presents an approach to predicting suicidal ideation in social media data. The study highlights the potential of machine learning techniques, including TF-IDF and vector decomposition, for analyzing social media data related to suicidal ideation. Valeria et al. [35] detect suicidal intent in Spanish language social networks using machine learning techniques. The authors collect data from Twitter and apply various machine learning algorithms, including decision trees, Naïve Bayes, and random forest, to classify tweets as either suicidal or non-suicidal based on TFIDF features. Ji et al. [4] presented a dataset for suicidal ideation as well as a method for detecting early suicide ideation on Twitter and Reddit. Michael et al. [5] presented a machine learning-based and deep learning-based classification approach to detect suicidal ideation on Reddit social media.

C. DEEP LEARNING

Deep learning has proven a huge success in a variety of areas, including computer vision, NLP, and medical diagnosis. It is also an important method for automatic suicidal ideation identification and suicide prevention in the field of suicide research. Mirtaheri et al. [36] proposed a novel learning model aimed at detecting suicidal ideation from social media posts, specifically from platforms like Twitter and Reddit. The model leverages a combination of natural language processing and advanced deep learning techniques, including an ensemble LSTM-TCN model enhanced with a self-attention mechanism. Boonyarat et al. [37] explored the application of advanced natural language processing (NLP) techniques, particularly using BERT models, to identify suicidal ideation and emotional distress in Thai social media posts during the COVID-19 pandemic. The study introduces a manually annotated dataset of 2,400 Thai tweets. Ghosh et al. [38] proposed a lightweight and robust method for detecting depressive texts on social media. The authors utilized an attention-based bidirectional LSTM combined with a

CNN-based model. Additionally, some studies have focused on applying SVM-based approaches for similar tasks. Jawad et al. [39] introduced a novel deep learning model with a novel optimization approach to predicting depression based on internet posts. The study's goal is to identify suicidal thought patterns in individuals and to develop a deep learning method for diagnosing depression using a constrained collection of features. Aldhyani et al. [40] present an approach to identifying suicidal tendencies on social media platforms by proposing a hybrid deep-learning model. The model achieves 95% accuracy in detecting suicidal ideation from social media posts. The model utilizes Word2Vec and CNN-BiLSTM for feature extraction and classification. Renjith et al. [41] present a hybrid LSTM-Attention-CNN model for analysing social media material for suicidal thoughts. Tadesse et al. [5] utilized an LSTM CNN model with word embeddings as input to detect suicidal ideation on Reddit.

D. FEATURE SELECTION

Feature selection is the process of removing irrelevant and duplicated information from a feature set to increase the accuracy of classifiers. Improved classification effectiveness requires feature selection [42]. Sameen et al. [43] used Random forests-based feature selection (RFFS) to improve text classification accuracy. Bidi et al. [44] presented a methodology to improve text classification accuracy using feature reduction approaches represented in the GA.

Overall, these studies have some limitations, such as (1) focusing solely on the level of words without regard for context meaning. (2) Create a few feature sets and feed them into the learning algorithm. (3) Many features are extracted. However, all of these are used as inputs to the learning algorithm during the classification process. Which includes some superfluous and redundant elements. These features will fool the learning algorithm.

III. PROPOSED FRAMEWORK

The primary objective of this paper is to introduce a novel framework that utilizes an intelligent approach for detecting suicide ideation from textual data. The proposed framework focuses on text classification and enhances the performance of language analysis in suicide detection by leveraging machine learning and NLP techniques. The framework aims to swiftly identify suicidal ideation in text by utilizing context-related features and feature selection methods that minimize computation and time complexity. Fig. 1 illustrates the workflow of the framework, which consists of several sequential phases. The data collection phase includes the data collection method and a detailed description of the dataset. The data preprocessing phase involves cleaning and preparing the text for the feature engineering phase. The feature engineering phase comprises two sub-phases: feature extraction, where context-related features (referred to as linguistic features) are extracted, and feature reduction, where relevant and informative features are selected as inputs for the classification phase. In the classification phase, machine learning algorithms are employed to detect suicidal thoughts based on the input features and analyze the results.

TABLE 1. Examples of suicide posts and non-suicide posts on reddit social media.

Suicidal Posts	Non-Suicidal Posts
I'll be dead, just you wait and see.	I bought shoes from a drug dealer once.
I think I'm ready for the long sleep.	Three men arrive at a forest.
I hate my parents so I'll kill myself.	You think microwave spying on you is bad.
My life has no purpose, I am broke.	Mike Tyson open the door to alcoholism?
I just have no will to live anymore.	Coffee shop and or brothel?

TABLE 2. Annotation rules of reddit posts.

Categories	Ruls
Suicidal Post	- Suicidal thoughts expressed - Suicidal ideation is included
Non-suicidal Post	- Formally debating suicide - Referring to the suicide of others - Not related to suicide

A. DATA COLLECTION PHASE

In order to detect suicide, classification models were trained using a dataset sourced from the Reddit social media platform. The dataset utilized in this study was compiled by Ji et al. [4] and consists of a collection of posts classified as either suicidal or non-suicidal. To ensure user privacy, unique IDs were employed instead of personal information. As users often engage in multiple subreddits, each group within the dataset contains a similar random number of posts from various topics. Our dataset comprises 3652 non-suicidal posts and 3549 posts discussing suicide, sourced from relatively large subreddits specifically dedicated to supporting individuals at risk. Suicidal posts were collected from the ''SuicideWatch''(SW) subreddit.[3] Popular subreddits[4] provided posts with no suicidal intent. Non-suicidal data collection is entirely user-generated content, with no news aggregation posts or administrators. Non-suicidal posts are generated from subred-dits that are thematically relevant to family and friends. Table 1 displays examples of both suicide and non-suicide attempts.

All the posts underwent a manual verification process to ensure accurate labeling. In Table 2, you can find our annotation rules and examples of the posts.

B. DATA PREPROCESSING PHASE

Data preprocessing plays a crucial role in all machine learning tasks. Once the dataset was collected, each post underwent preprocessing using NLP techniques before feature extraction. The Natural Language Toolkit (NLTK) and Spacy were employed to preprocess the dataset before proceeding to the training phase. Preprocessing involves a

[3]https://www.reddit.com/r/SuicideWatch/

[4]https://www.reddit.com/r/all/,https://www.reddit.com/r/popular/

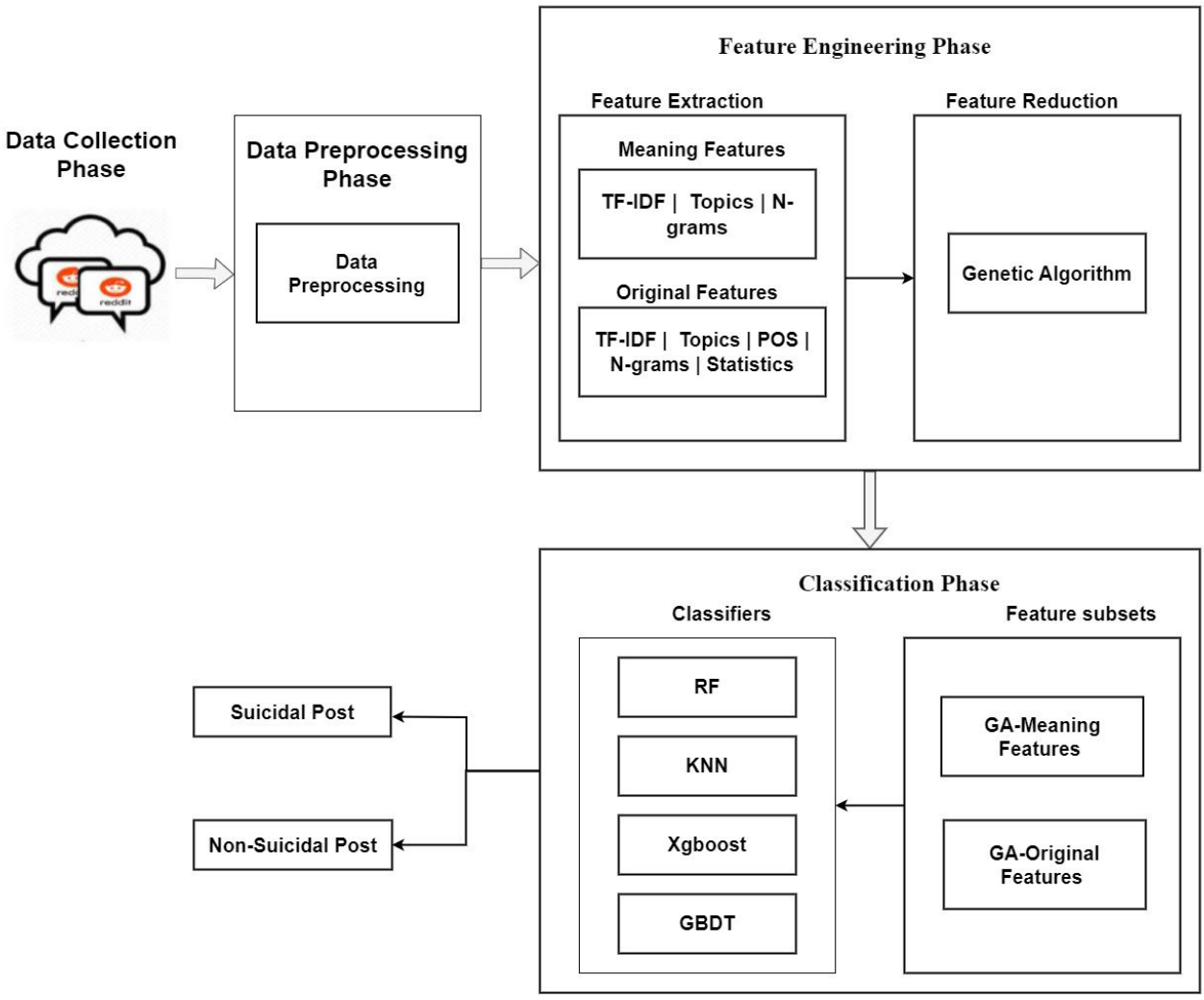


FIGURE 1. Proposed framework architecture.

series of filtering stages applied to Reddit posts, converting raw data into a format that can be comprehended by machine learning models.

- Case conversion. Convert text to lowercase.
- Tokenization. The posts were tokenized using words.
- Stripping HTML tags from the data.
- Removing accented characters. for example "Sómě Áccěntěd těxt" transformed into " Some Accented text".
- Expanding contraction. Firstly, a contraction map was created, that includes all possible contractions and their expansions, such as "haven't": "have not", "he'd": "he had / he would", "he'd've": "he would have", and so on. Then, using the contraction map, all the contractions in each post are compared and transformed into their expansion.
- Removing special characters. Any character except a-z, A-Z and 0-9 was removed.
- Removing repeated characters. The duplicate letters were removed in collabora-tion with the dictionary.
- Removing stop words. A custom stop-word list is created. Stop words from the NLTK and Spacy stop words lists were used, some stop words that could change the meaning of the sentence are left out, such as "not," "myself," "alone," "against," "nobody," 'still," and so on.
- Lemmatization. Where word endings cannot be dropped, resulting in meaningless word parts as stemming does.

Now, the data is ready for the feature engineering phase.

C. FEATURE ENGINEERING PHASE

Feature engineering refers to the process of selecting, transforming, extracting, combining, and manipulating raw data to generate the desired variables for analysis or predictive modeling. It is a fundamental step in the development of a

machine learning model. When executed effectively, feature engineering enhances the predictive capability of machine learning algorithms by creating features from raw data that facilitate the learning process. This phase is divided into two sub-phases: feature extraction and feature selection.

1) FEATURE EXTRACTION

Feature extraction is the process of deriving meaningful information by extracting features from a dataset. This reduces the data size to a manageable quantity for algorithms to process while preserving the original relationships and important information. In this phase, two feature sets were extracted. (1) Original features set, which includes all extracted features and encompasses five combined features: statistics, POS, TF-IDF, N-grams, and topic features. (2) Linguistic features set, which focus solely on meaningful and context-related features, comprising three combined features: TF-IDF, N-grams, and topic features. The two sets are the output of the feature extraction phase. These two sets were compared, and learning algorithms were tested on them.

1) Statistics features
   The length of user-generated posts may differ, with some being shorter and consisting of simple sentences, while others are longer and include complex paragraphs. After segmenting and tokenizing the text, the number of words, tokens, characters, sentences, and paragraphs is calculated as a statistical feature.
2) Part of speech (POS) features
   POS [45] are essential features extracted from text content in natural language processing. POS tagging involves assigning grammatical labels to words in a sentence, such as nouns, verbs, adjectives, adverbs, pronouns, conjunctions, and more. By extracting POS as features, we gain valuable insights into the syntactic structure and grammatical properties of the text. This information can be leveraged for various NLP tasks, including sentiment analysis, named entity recognition, text classification, and even suicidal ideation detection. Each user post was tagged and parsed, and the number of times each category appeared in the user content was simply counted.
3) Linguistic features
   Language usage patterns are analyzed to extract linguistic features, including word choice, topics of interest, and sentence structure. One prevalent technique for identifying such features involves converting the user's expressed text content into a psychological dictionary and tracking the frequency of words within each word class.

   a) Topic features
      Topic features in the text refer to the underlying themes or subjects that are present within a given piece of written or spoken content. These features represent the main ideas or concepts that are being discussed or explored within the text. Identifying and analyzing topic features can be beneficial in various fields, such as NLP, information retrieval, text mining, and text classification. Latent Dirichlet Allocation (LDA) [46] is a powerful technique for extracting meaningful features from text data, and it can be particularly useful in detecting suicidal ideations. LDA is a probabilistic model that treats each document as a mixture of topics, with each topic being a distribution of words. By analyzing a collection of documents, LDA can identify the underlying topics and the words associated with each topic.

   b) Word frequency features (TF-IDF)
      TF-IDF (Term Frequency-Inverse Document Frequency) is a widely used feature in text classification tasks. It represents the importance of a term in a document relative to a collection of documents. TF-IDF takes into account both the frequency of a term within a document (TF) and its rarity across the entire document collection (IDF). The TF component measures how often a term appears in a document. It assigns higher weights to terms that occur frequently within a document, assuming that those terms are more representative of the document's content. On the other hand, the IDF component calculates the rarity of a term by considering its occurrence across the entire document collection. Terms that appear frequently across many documents are given lower weights, as they likely do not carry significant discriminatory power. TF-IDF is employed to evaluate the importance of words within user posts related to suicidal and non-suicidal content. The significance of a word is determined by the TF-IDF equations, which are outlined below.
      TF measures the frequency of a term t within a document d. It is calculated using the following formula:

$$tf(t,d) = \frac{C_{t,d}}{\sum_k C_{t,d}} \tag{1}$$

      $C_{t,d}$ denotes the frequency of term t within document d, while $\sum_k C_{t,d}$ represents the total count of terms in document d. The inverse document frequency (idf) is computed using the following formula.

$$idf(t,D) = log\frac{D}{d_t} + 1 \tag{2}$$

      D refers to the total number of posts in the dataset, while d_t represents the number of posts in the dataset that contain the term t. The tfidf value is obtained by multiplying Eq 1 with Eq 2.

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D) \tag{3}$$

c) Ngram features
The post is represented as a bag of words in BoW, with grammar and even word order being ignored but word multiplicity being kept. The disadvantage of BoW is that the order of the words is lost, which means that valuable info about the user's psychological state is lost. This problem is solved by the N-gram model [47]. N-grams features are a type of feature representation in natural language processing that capture the sequential relationships between adjacent words or characters in a text. N-grams are formed by dividing the text into contiguous sequences of n words or characters. By considering these n-gram sequences, models can capture important contextual information, such as phrases or language patterns, which can be useful in various tasks like language modeling, sentiment analysis, and text classification. The choice of the value of "n" determines the length of the n-gram and the level of contextual information captured. In the proposed framework, a trigram model based on N-Gram is employed to forecast sequences of suicidal words in user posts, whether they exhibit suicidal or non-suicidal tendencies. One of the most significant limitations of earlier studies in detecting suicidal ideation is that they were unconcerned about the sentence's intended meaning. Previous research only looked at individual words in a user's post; however, what about having the words next to each other? What about the arrangement of the consecutive words? Previous works suppose that the presence of the phrase "I want to kill myself" is the same as the existence of the paragraph "I do not want to kill myself" because it is unconcerned with the words next to each word or the arrangement of the words.

The details of how the feature extraction phase is implemented are elucidated in the pseudocode presented in Algorithm 1. In this algorithm, five distinct sets of features are extracted from each post and amalgamated into two sets, referred to as the original set and the linguistic set. To enhance clarity, certain notations are furnished in Table 3 to facilitate comprehension of the algorithm.

Now, let's delve into the complexity analysis. To assess the algorithm's complexity, we calculate the CPU time needed to execute each statement, expressed in terms of Big O notation. Certainly. Various notations are used to describe the time and space complexities of different operations in the feature extraction algorithm. Here's an explanation of the notations used in the table 4.

The time complexity for each feature extraction is given as $O(N \cdot X)$, where X is the respective complexity for that feature extraction. The space complexity is also represented similarly. The total time complexity for the entire feature extraction algorithm is then given by $O(N \cdot (M + P + Q + R + S))$, representing the sum of the complexities for each feature extraction operation multiplied by the number of records in the dataset. Table 5 illustrates the complexity analysis for each step in the feature extraction phase as well as the overall complexity analysis of the feature extraction algorithm 1.

TABLE 3. Notation of algorithm 1.

Symbol	Meaning
docs	the posts of users in the dataset $\{ \text{doc}_1, doc_2, ... doc_n \}$
org-set	a set that concatenates all the extracted features
ling-set	a set that concatenates only the linguistic features
getTFIDF	function to extract TFIDF features
getTopics	function to extract topic features
getNgram	function to extract ngram features
getPOS	function to extract POS features
getStatistcs	function to extract statistical features

Algorithm 1 Feature Extraction Algorithm

```
0: procedure FeatureExtraction(docs)
1: original_set_features ← []
2: linguistic_set_features ← []
2:    for each doc in docs do
3: tfidf_features ← getTFIDF(doc)
4: topics_features ← getTopics(doc)
5: ngram_features ← getNgram(doc)
6: pos_features ← getPOS(doc)
7: statistics_features ← getStatistics(doc)
8: original_set_features.append(tfidf_features)
9: original_set_features.append(topics_features)
10: original_set_features.append(ngram_features)
11: original_set_features.append(pos_features)
12: original_set_features.append(statistics_features)
13: linguistic_set_features.append(tfidf_features)
14: linguistic_set_features.append(topics_features)
15: linguistic_set_features.append(ngram_features)
15:    end for
16: return original_set_features,linguistic_set_features
16: end procedure=0
```

TABLE 4. Notations used in complexity analysis of alogorithm 1.

Symbol	Description
N	Number of records or posts in the dataset
M	Complexity of TFIDF extraction for one record
P	Complexity of Topics extraction for one record
Q	Complexity of Ngram extraction for one record
R	Complexity of POS extraction for one record
S	Complexity of Statistics extraction for one record

2) FEATURE SELECTION

One of the contributions of this study is integrating feature selection approaches into suicide ideation detection and

TABLE 5. Complexity analysis of algorithm 1.

Feature	Time Complexity	Space Complexity
TFIDF Extraction	$O(N \cdot M)$	$O(N \cdot M)$
Topics Extraction	$O(N \cdot P)$	$O(N \cdot P)$
Ngram Extraction	$O(N \cdot Q)$	$O(N \cdot Q)$
POS Extraction	$O(N \cdot R)$	$O(N \cdot R)$
Statistics Extraction	$O(N \cdot S)$	$O(N \cdot S)$
Total Time Complexity		$O(N \cdot (M + P + Q + R + S))$

evaluating their influence on the suicide detection process. Feature selection is a crucial step in machine learning and data analysis, aimed at identifying the most relevant and informative variables or attributes from a given dataset. It involves the process of selecting a subset of features that contribute significantly to the predictive power of a model while excluding irrelevant or redundant ones.

Effective feature selection not only reduces the dimensionality of the data, which can improve computational efficiency, but also helps in mitigating the risk of overfitting and enhancing the interpretability of the model. The ultimate goal of feature selection is to identify a concise and representative set of features that capture the essential information in the data, leading to more accurate and robust models.

Users' posts often consist of hundreds or even thousands of words, resulting in high-dimensional feature spaces. However, within this abundance of words, many features may be noisy or redundant, lacking meaningful information for the learning algorithm. This can potentially mislead the algorithm and hinder accurate predictions.

Therefore, it becomes essential to employ effective feature selection techniques to identify the most relevant and informative features, reducing the dimensionality of the dataset and improving the overall performance and interpretability of the learning algorithm. A wrapper feature selection method is used to solve this problem. The genetic algorithm (GA) is used for feature selection because it is relatively faster for higher dimensions.

Holland and his colleagues introduced the genetic algorithm (GA) in 1960, drawing inspiration from the theory of evolution [48]. GA is an evolutionary algorithm that produces optimized individuals by iteratively refining a set of initial solutions. The primary components of GA are the three fundamental genetic operators: selection, crossover, and mutation [49]. Fig. 2 shows a simplified flowchart of the genetic algorithm.

- Initial population: In the GA, each potential solution is represented as a string of binary or real-valued numbers, which is called a chromosome or initial population, and each chromosome consists of attributes called "genes". To determine the appropriate population size, it is not ideal to have a population that is either too large or too small. Therefore, the most effective approach is to use a trial-and-error method [50].
- Selection: The goal of selection is to choose potential chromosomes from the present population to be the parents for the crossover operation. In the selection process, a roulette wheel selection approach is used to choose the parent [51]. The fitness values computed from the chromosomes are first converted into probability in this scheme. The parent is then chosen at random by the roulette wheel based on the probability values. A chromosome with a higher fitness value is more likely to become a parent [52]. The probability of choosing a parent can be expressed as:

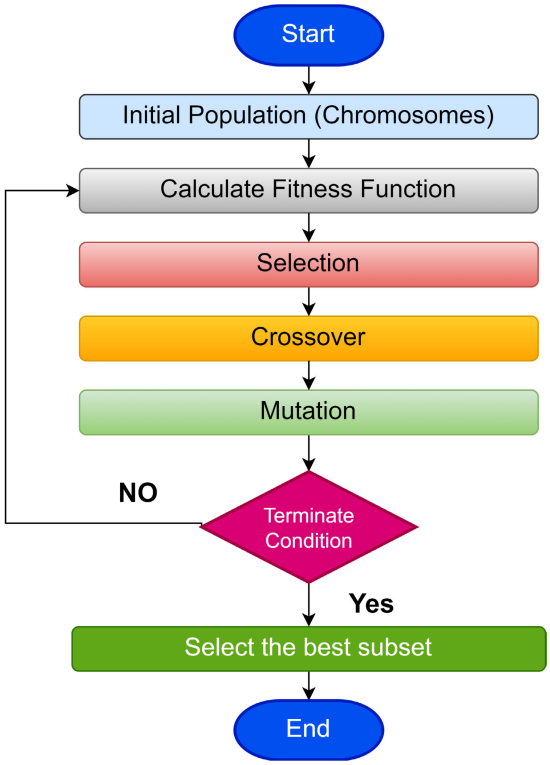


FIGURE 2. Flow chart of a genetic algorithm.

$$P(C_i) = \frac{1 - F(C_i)}{\sum_{i=1}^{N} 1 - F(C_i)} \tag{4}$$

where C_i represents the chromosome (user post), i is the chromosome order in the population, $F(C_i)$ is the fitness function, $P(C_i)$ is the probability of selecting individual C_i for reproduction, and N represents the population size.

For greater clarity on the feature selection process in the study, the following provides a detailed explanation of the fitness function used in the GA, along with the criteria and metrics employed during the GA's evaluation stage:

- **Input Parametrs**
  - **individual:** A binary vector representing the selection of features (1 for selected, 0 for not selected).
  - **X:** The dataset containing all features.

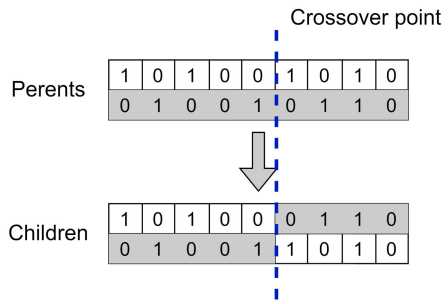


FIGURE 3. Crossover operation example.

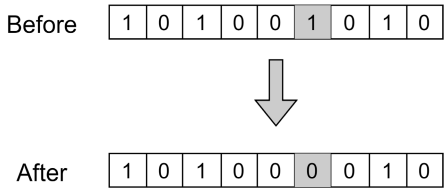


FIGURE 4. Mutation operation example.

- **Y:** The target vector with the labels for classification.

– **Fitness Calculation Process**

- The code checks if the individual vector is not entirely zeros (i.e., at least one feature is selected).
- It identifies the indices of the features that are not selected (where individual elements are 0).
- It creates a new dataset X_{parsed} by removing the unselected features.
- It further processes X_{parsed} to one-hot encode any categorical features, resulting in X_{subset}.

– **Model Training, Evaluation, and Used Criteria**

- A logistic regression classifier is instantiated.
- 5-fold cross-validation is performed using the classifier on the subset of features.
- The accuracy scores from the cross-validation are averaged to get a single fitness value.

- Crossover: crossover is performed between two selected parents to generate children (new solutions). At this point, a crossover rate (CR) will determine the likelihood that two parents swapped their genes. We use the single-point crossover method, which is the most commonly used in GA [52]. Fig. 3 depicts a crossover operation example. As can be seen, two parents traded their genetic code in order to have two offspring, with the children inheriting genes from both parents [52].
- Mutation: It seeks to modify the genes of freshly created children (new chromosomes) in nature. The chromosome's genes are altered from bit "1" to bit "0" or from bit "0" to bit "1" depending on the mutation rate (MR) [53]. Fig. 4 depicts a mutation operation example. Where a chromosomal gene was chosen at random and altered from bit "1" to bit "0."

TABLE 6. Symbols used in genetic algorithm pseudocode 2.

Symbol	Description
N	Population size
G	Number of generations
initializePopulation()	Initialization function
`evaluatePopulation()`	Fitness evaluation function
`selectParents()`	Selection function
`crossover()`	Crossover function
`mutate()`	Mutation function
`replacePopulation()`	Replacement function
`getBestIndividual()`	Function to retrieve the best individual

The details of how the feature selection phase is implemented are elucidated in the pseudocode presented in Algorithm 2. To enhance clarity, certain notations are furnished in Table 6 to facilitate comprehension of the algorithm.

Algorithm 2 Genetic Algorithm Pseudocode

```
Procedure GeneticAlgorithm(N, G)
initializePopulation(N)
for generation in range(G) do
evaluatePopulation()
selectParents()
crossover()
mutate()
replacePopulation()
end for
return getBestIndividual()
end procedure=0
```

Now, let's delve into the complexity analysis. To assess the algorithm's complexity, we calculate the CPU time needed to execute each statement, expressed in terms of Big O notation. Certainly. Various notations are used to describe the time and space complexities of different operations in the feature selection algorithm. Here's an explanation of the notations used in Table 7.

TABLE 7. Notations used in genetic algorithm complexity analysis 2.

Symbol	Description
N	Population size
G	Number of generations
F	Complexity of fitness evaluation
S	Complexity of selection
C	Complexity of crossover
M	Complexity of mutation
R	Complexity of replacement

Table 8 provides the time complexity for each step in the genetic algorithm and the overall time complexity.

During the feature selection phase, the GA is employed with an initial population of 200 chromosomes, running for 100 generations (iterations) on two feature sets: the original feature set and the linguistic feature set obtained from the feature extraction phase. After applying the GA,

TABLE 8. Complexity analysis of genetic algorithm 2.

Step	Time Complexity
Initialization	$O(N)$
Evaluation	$O(N \cdot F)$
Selection	$O(N \cdot S)$
Crossover	$O(N \cdot C)$
Mutation	$O(N \cdot M)$
Replacement	$O(N \cdot R)$
Total Time Complexity	$O(G \cdot (N + F + S + C + M + R))$

two new feature sets are generated: GA-original features and GA-linguistic features.

- The original feature set: It comprises a collection of all the extracted features, including statistical features, TF-IDF features, topic features, n-gram features, and POS features. Table 9 displays the features that were extracted and included in the original feature set.

TABLE 9. Original features set details.

Statistics	POS	TF-IDF	Topics	N-gram	Total Features
8	32	50	10	100	200

- The Linguistic Features Set (Meaning Features): It is a set of features that capture the context and meaning in the proposed framework, including n-gram features, TF-IDF features, and topic features. The n-gram features provide insight into the linguistic patterns within users' posts on Reddit. The TF-IDF features measure the importance of each word within the posts, taking into account its frequency in the corpus. Finally, the topic features uncover latent themes and subjects present in the users' posts on Reddit. Table 10 shows the features that were extracted and included in the Linguistic feature set.

TABLE 10. Linguistic features set details.

TF-IDF	Topics	N-gram	Total Features
50	10	100	160

- The GA-original features. It represents the selected features from the original feature set determined by the GA. Table 11 displays details on GA-original features.
- The GA-linguistic features represent the selected features from the linguistic feature set determined by the GA. Table 12 displays details of GA-Linguistic features.

TABLE 11. Details of GA-original features.

Category	Total Features	Selected Features by GA
Statistics	8	1
POS	32	9
TF-IDF	50	22
Topics	10	9
N-gram	100	45
Total Features	200	86

TABLE 12. Details of GA-linguistic features.

Category	Total Features	Selected Features by GA
TF-IDF	50	13
Topics	10	7
N-gram	100	39
Total Features	160	59

As shown in Table 11, GA chose most of the selected features from linguistic (context-related and meaningful) features. So, we focused on linguistic features, which yield a better result in determining whether the text contains suicidal ideation.

D. CLASSIFICATION PHASE

The classification of a suicidal ideation detection framework involves multiple stages, one of which includes feature selection using genetic algorithms (GA). After applying GA to identify the best features from the dataset, these selected features are utilized as inputs for the classification process. To evaluate the performance of the framework, four classifiers, namely Random Forest (RF) [54], k-Nearest Neighbors (KNN) [55], XGBoost (XGB) [56], and Gradient Boosting Decision Trees (GBDT) [57], are employed. Each classifier utilizes the selected features to learn patterns and make predictions regarding suicidal ideation. By comparing the results obtained from these four classifiers, The tree ensemble methods RF, GBDT, and XGB use decision trees as base classifiers and produce a type of committee that surpasses any single base classifier. Another prominent classifier for mental health assessments is KNN [58]. Each algorithm is applied to four different subsets: (1) the original features set, which consists of all the extracted features; (2) the meaning features set, which exclusively includes context-related features such as tf-idf, topics, and N-gram; (3) the GA-original features, which are selected from the original features set using GA; and (4) the GA-meaning features, which are selected from the meaning features set using GA. Choosing the best hyperparameters and their values is crucial for optimizing model performance and ensuring that a classifier generalizes well to unseen data. Properly tuned hyperparameters can significantly enhance the model's accuracy, robustness, and efficiency. By adjusting these parameters, we balance the trade-offs between model complexity and generalization, aiming to achieve the best performance while avoiding issues such as overfitting or underfitting. This careful tuning process is essential for maximizing the effectiveness of the machine learning models used in our study. The best performance in our research was achieved using the hyperparameters shown in Table 13.

E. EVALUATION METRICS

To evaluate the performance of the proposed framework, various metrics for the evaluation process are utilized, including accuracy, precision, recall, and F1 score [59]. These metrics rely on the information provided by the confusion

TABLE 13. Model parameters and their effects on performance.

Classifier	Parameter	Value	Description
RF	n_estimators	100	Number of trees in the forest; more trees reduce variance and improve accuracy.
	max_depth	8	Maximum depth of each tree; limits complexity to prevent overfitting.
	random_state	42	Seed for random number generation; ensures reproducibility of results.
	n_jobs	-1	Uses all CPU cores for computation; speeds up training and prediction.
KNN	n_neighbors	200	Number of neighbors for predictions; a higher value smooths decision boundaries and improves generalization.
	weights	uniform	Equal influence of neighbors; simplifies the model.
	algorithm	auto	Automatically selects the best search algorithm for efficiency.
GBDT	n_estimators	50	Number of boosting stages; more stages enhance learning but risk overfitting.
	max_depth	8	Maximum depth of trees; controls complexity to prevent overfitting.
	random_state	42	Ensures consistent results across runs.
XGBoost	max_depth	8	Limits tree depth to balance learning complex patterns and overfitting.
	eta	0.2	Learning rate; controls convergence speed and accuracy.
	objective	multi:softprob	Multi-class classification objective; predicts class probabilities.
	eval_metric	mlogloss	Performance metric using logarithmic loss; measures precision in multi-class classification.
	num_boost_round	10	Number of boosting rounds; balances learning capacity and risk of overfitting.

matrix, which includes the number of true negatives (TN), false negatives (FN), true positives (TP), and false positives (FP).

1) ACCURACY (ACC)

is a metric that assesses a model's ability to correctly classify data. It quantifies the number of accurately categorized negative and positive samples, as shown in Eq 5.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

2) PRECISION (P)

is a metric that measures the performance of a model in predicting data samples belonging to the positive class. It calculates the percentage of positive samples that were accurately predicted, as depicted in Eq 6.

$$P = \frac{TP}{TP + FP} \tag{6}$$

3) RECALL (R)

It is a metric that is also known as sensitivity. It quantifies the percentage of positive samples correctly predicted out of all the samples in the actual class, as expressed in Eq 7.

$$P = \frac{TP}{TP + FN} \tag{7}$$

4) F1-SCORE (F1)

is a weighted average of precision and recall. It considers both false negative and false positive results, as outlined in Eq 8

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{8}$$

IV. RESULTS AND DISCUSSION

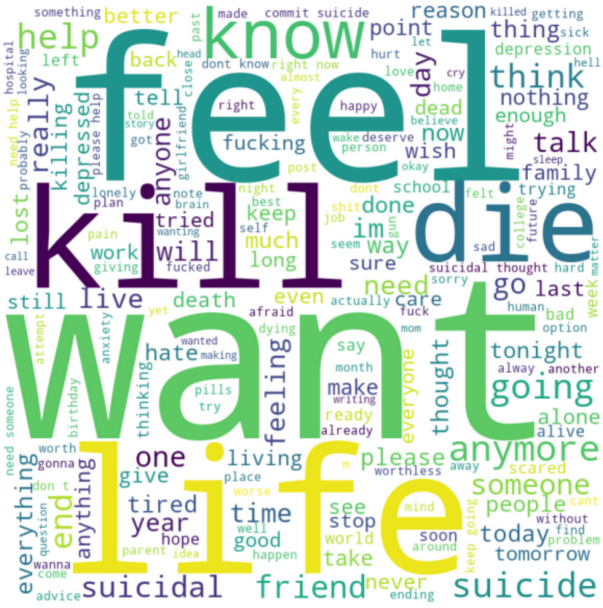
Section IV-A provides the tools and software used in this experiment.Section IV-B provides the results of data language analysis. The results obtained from the proposed framework are presented in Section IV-C. Additionally, section IV-D delves into a comparative analysis with previous studies conducted on the same dataset, providing insights and discussions on the findings.

A. EXPERIMENTAL SETUP

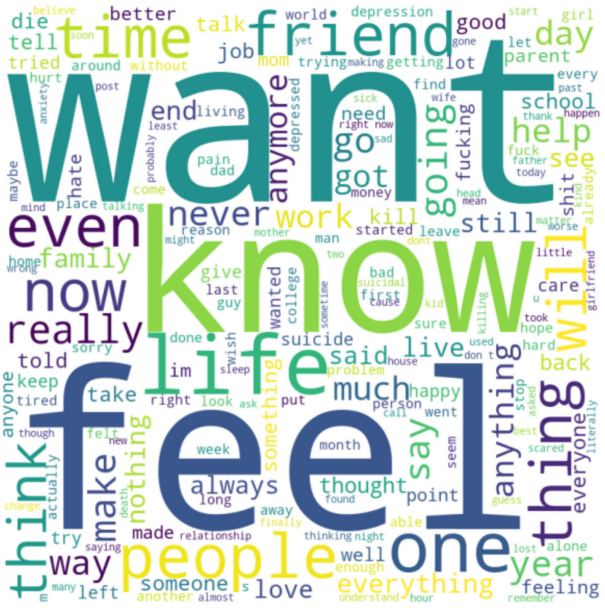
The experiments were conducted on a laptop with an Intel Core i3 CPU, 12 GB of RAM, and the 64-bit Windows 10 operating system. The NLTK and Spacy libraries were utilized for data cleaning and feature extraction, while the sklearn library was employed for implementing classification models and feature selection algorithms in Python.

B. DATA ANALYSIS RESULTS

An analysis was conducted on the dataset to identify instances of suicidal thoughts and explore variations in the vocabulary used. Word clouds were employed to generate visual representations of the data. Fig5 (a) and Fig 5 (b) show which words are more prevalent in suicide-related and



(a) Suicidal Posts



(b) Non-Suicidal Posts

FIGURE 5. Word cloud visulazation.

non-suicide-related texts or posts. As illustrated in Fig. 5 (a), suicide posts frequently use words like "die", "life", "kill", "suicide", and "suicidal", which provides a direct indication of users contemplating suicide. Words expressing intentions or feelings, such as "want", "think", "know", and "feel," are frequently used. Suicidal posts, for example, include phrases such as "I'll be dead, just you wait and see.", "My life is meaningless," and "I think I'm ready for the long sleep." as shown in Table 1. As illustrated in Fig 5 (b), non-suicide posts frequently use words like "friend", "feel", "people" and "life", which do not provide a direct indication of users contemplating suicide.

C. EXPERIMENTAL RESULTS

This paper performed a comparative analysis of various classification algorithms using four different feature sets. The classification methods evaluated were RF, KNN, GBDT, and XGBoost. The study employed two methods for splitting the data: the train-test split and k-fold cross-validation. The results section will compare the performance of the classifiers tested using both methods to provide a comprehensive evaluation. The classifiers performance was evaluated based on four different evaluation metrics: accuracy, precision, recall, and F1 score. The testing time of the test proportion was also calculated.

In the first experiment, the train test split method is used, 70% of the dataset was used for training the classification models, while the remaining 30% was reserved for evaluating the detection model's performance. Table 14 encapsulates the performance results of the proposed framework across the used classifiers, revealing the impact of

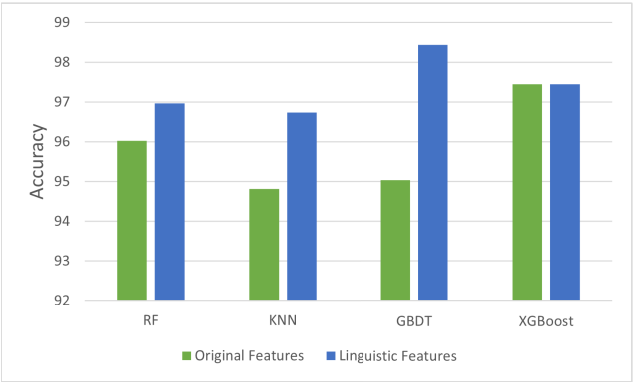


FIGURE 6. Train test split results without feature selection.

incorporating linguistic features alongside the original feature set without employing feature selection techniques. Two sets of features were considered for each classifier: the original feature set comprising 200 features and an augmented set encompassing linguistic features, which comprises 160 features.

In terms of overall accuracy, the classifiers demonstrated robust performance. Notably, when utilizing the original features, RF achieved an accuracy of 96.03%, while KNN and GBDT achieved 94.81% and 95.04%, respectively. The addition of linguistic features yielded improvements across all classifiers, with the most substantial enhancements observed in the KNN and GBDT classifiers, resulting in accuracies of 96.97% and 98.44%, respectively. Precision, recall, and F1-score metrics further emphasize the effectiveness of

TABLE 14. Train test split results without feature selection.

Classifier	Features Set	Accuracy	Precision	Recall	F1-Score	Testing time (ms)
RF	Original Features (200)	96.03	95.41	96.98	96.19	181
	Linguistic Features (160)	96.97	95.49	98.81	97.12	115
KNN	Original Features (200)	94.81	90.86	100.0	95.21	1300
	Linguistic Features (160)	96.74	96.29	97.44	96.86	646
GBDT	Original Features (200)	95.04	91.24	100.0	95.42	112
	Linguistic Features (160)	98.44	97.49	99.54	98.51	78
XGB	Original Features (200)	97.45	95.29	100.0	97.59	146
	Linguistic Features (160)	97.45	95.29	100.0	97.59	115

incorporating linguistic features. Across various classifiers, the linguistic feature set consistently contributed to improved precision and recall, resulting in higher F1-scores. This enhancement is particularly evident in the KNN and GBDT, where precision, recall, and F1-score approached or achieved 100% with the inclusion of linguistic features. Fig. 6 illustrates the impact of utilizing linguistic features instead of original features in detecting suicidal ideation. Additionally, the testing time (measured in milliseconds) provides insights into the computational efficiency of the classifiers. While there are variations in testing times across classifiers, the linguistic feature set generally leads to a reduction in testing time. This reduction signifies the potential computational efficiency gained through the inclusion of linguistic features, making the proposed framework not only more accurate but also computationally more efficient.

The results presented in Table 15 offer valuable insights into the impact of feature selection on the performance of the proposed framework across various classifiers. Two distinct feature sets were considered: GA-Original Features and GA-Linguistic Features. The GA-Original Features set comprises 86 features selected through a genetic algorithm from the original features set, while the GA-Linguistic Features set incorporates 59 features, representing a subset of linguistically derived features chosen by the same genetic algorithm from the linguistic features set.

Across classifiers, the GA-Linguistic Features consistently demonstrated improvements in accuracy, precision, recall, and F1-score metrics when compared to the GA-Original Features set. RF, KNN, and XGB classifiers, in particular, exhibited enhanced performance with the inclusion of linguistic features. The GA-Linguistic Features set, with its reduced dimensionality, yielded a more discriminative and informative subset, leading to increased accuracy and more balanced precision-recall trade-offs.

The testing time results reveal noteworthy reductions in computational costs associated with the GA-Linguistic Features set. Fig. 7 also demonstrates the impact of using the GA as a feature selection technique across both the original and linguistic feature sets. The reduced feature set led to quicker testing times across all classifiers, with the most significant improvements observed in the GBDT classifier, where testing time was substantially reduced from 26 ms to 12 ms. This reduction underscores the efficiency gains achieved through the intelligent selection of linguistically relevant features, further affirming the practical advantages of feature selection in enhancing both accuracy and computational efficiency.

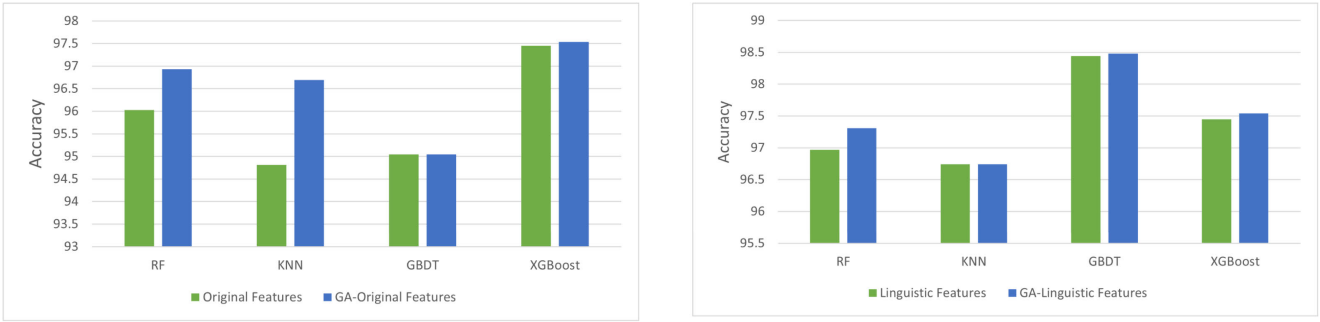
In the second experiment, to enhance the robustness and reliability of our models, we conducted a 5-fold cross-validation methodology. This approach is essential in machine learning as it helps mitigate the risk of overfitting and provides a more accurate assessment of model performance. By dividing the dataset into five subsets, we trained the models on four parts while validating them on the remaining part, iterating this process across all subsets. This not only ensures that every data point is used for both training and validation but also allows us to obtain a more generalized performance metric across different splits of the data.

The results presented in Table 16, obtained using 5-fold cross-validation without a feature selection phase, demonstrate the effectiveness of using linguistic features compared to original features across various machine learning algorithms. When utilizing the linguistic features set, which consists of 160 features specifically designed to capture the context and meaning of the data, we observe a significant improvement in the performance of all classifiers. The RF classifier achieves the highest accuracy of 97.79% using linguistic features, outperforming the original 200-feature set by a considerable margin. Similarly, the KNN and XGBoost classifiers also show substantial improvements in accuracy, precision, recall, and F1-score when employing the linguistic features. It is noteworthy that the performance of all classifiers, including GBDT, is consistently better when using the linguistic features set compared to the original features set as shown in Fig. 8. This finding highlights the importance of selecting features that are closely related to the context and meaning of the data, as they provide a more informative representation for the classifiers to learn from, ultimately leading to enhanced classification accuracy and overall model performance.

In Addition, Table 17 shows the results using a 5-fold cross-validation methodology along with GA as a feature selection technique, it is ovserved that the performance of the classifiers is significantly enhanced when employing GA to select the most important features. The results indicate that the GA-Linguistic Features set achieves the best performance, RF classifier attaining an impressive accuracy of 98.92%, showcasing the effectiveness of this approach.

TABLE 15. Train test split results with feature selection.

Classifier	Features Set	Accuracy	Precision	Recall	F1-Score	testing time (ms)
RF	GA-Original Features (86)	96.93	95.65	98.53	97.07	71
	GA-Linguistic Features (59)	97.31	95.68	99.27	97.44	58
KNN	GA-Original Features (86)	96.69	96.37	97.25	96.81	615
	GA-Linguistic Features (59)	96.74	96.46	97.25	96.85	585
GBDT	GA-Original Features (86)	95.04	91.24	100.0	95.42	26
	GA-Linguistic Features (59)	98.48	97.58	99.54	98.55	12
XGB	GA-Original Features (86)	97.54	95.54	99.90	97.67	85
	GA-Linguistic Features (59)	97.54	95.29	100.0	97.59	52



(a) Original Features

(b) Linguistic Features

FIGURE 7. Train test split results with feature selection.

TABLE 16. 5-fold cross validation results without feature selection.

Classifier	Features Set	Accuracy	Precision	Recall	F1-Score	Testing Time (s)
RF	Original Features (200)	95.81	96.13	95.81	95.80	2.70
	Linguistic Features (160)	97.79	97.88	97.79	97.79	1.64
KNN	Original Features (200)	95.21	95.52	95.21	95.20	1.91
	Linguistic Features (160)	97.68	97.76	97.68	97.68	1.64
GBDT	Original Features (200)	94.94	95.41	94.94	94.93	3.09
	Linguistic Features (160)	94.94	95.41	94.94	94.93	2.52
XGBoost	Original Features (200)	94.89	95.34	94.89	94.87	0.11
	Linguistic Features (160)	97.59	97.69	97.59	97.59	0.08

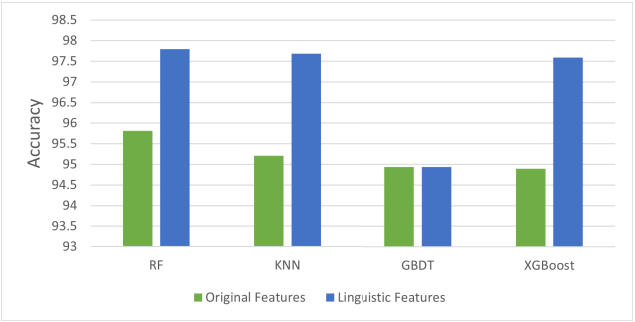


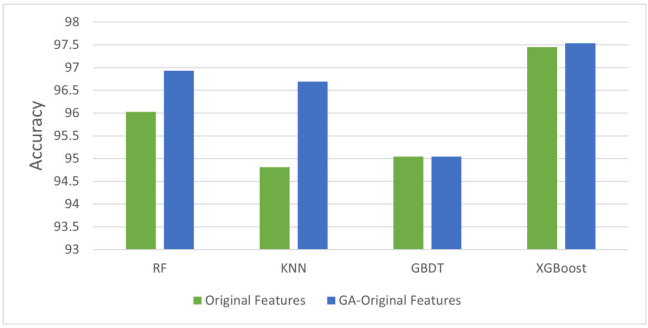
FIGURE 8. 5-fold results without feature selection.

This demonstrates that the combination of linguistic features, which inherently capture the latent meaning of user posts, and the GA for feature selection leads to superior results as shown in Fig. 9. In contrast, the classifiers using GA-Original Features generally exhibit lower performance metrics, reinforcing the notion that selecting the best features from the linguistic set is crucial for maximizing accuracy. Furthermore, the GA-Linguistic Features not only deliver the highest accuracy but also maintain lower testing times, indicating a more efficient and effective method for classification. This synergy between linguistic features and GA-based selection highlights the optimal strategy for achieving accurate results while minimizing complexity in the testing process

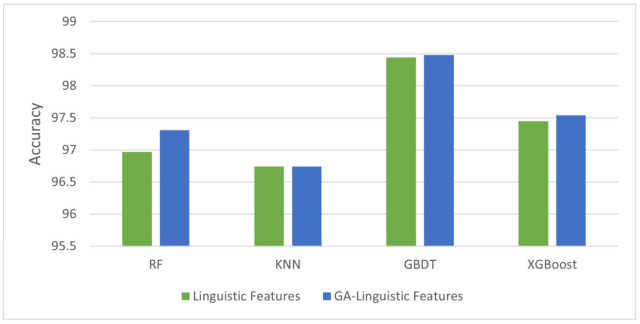
Based on the previous findings, it is evident that employing linguistic features combined with genetic selection yields the highest performance in both the train-test split and 5-fold cross-validation methodologies. Comparing the results obtained from the traditional train-test split methodology with those achieved through 5-fold cross-validation, it is evident that the latter provides a more robust and reliable assessment of model performance. The 5-fold cross-validation approach yielded the best performance, with the RF achieving an impressive accuracy of 98.92%. This significant improvement underscores the advantages of cross-validation in mitigating overfitting and ensuring that the model generalizes well to unseen data. Furthermore, the overall performance of all classifiers was enhanced when employing cross-validation, demonstrating its effectiveness

TABLE 17. 5-fold cross validation results with feature selection.

Classifier	Features Set	Accuracy	Precision	Recall	F1-Score	Testing Time (s)
RF	GA-Original Features (86)	95.07	95.50	95.07	95.06	2.93
	GA-Linguistic Features (59)	98.92	98.94	98.92	98.92	1.64
KNN	GA-Original Features (86)	94.82	95.25	94.82	94.80	1.90
	GA-Linguistic Features (59)	97.66	97.75	97.66	97.66	1.58
GBDT	GA-Original Features (86)	97.93	98.01	97.93	97.93	1.41
	GA-Linguistic Features (59)	97.93	98.01	97.93	97.93	2.19
XGBoost	GA-Original Features (86)	97.82	97.90	97.82	97.82	0.06
	GA-Linguistic Features (59)	98.85	98.85	98.85	98.85	0.15

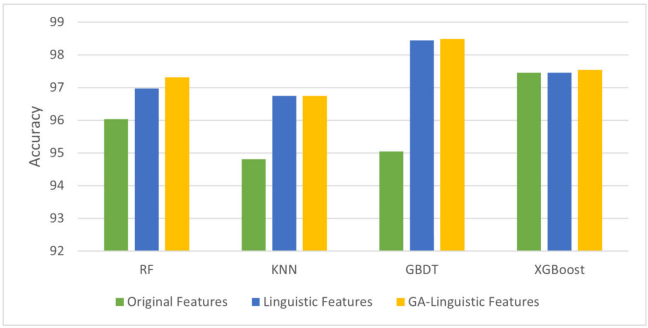


(a) Original Features

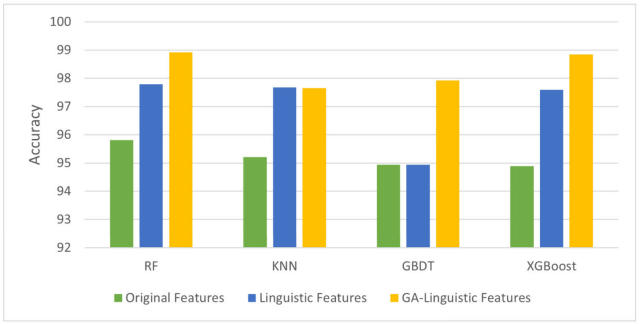


(b) Linguistic Features

FIGURE 9. 5-fold cross validation results with feature selection.



(a) Train test split



(b) 5-fold cross validation

FIGURE 10. Proposed framework result analysis.

in providing a comprehensive evaluation of model capabilities. This methodological rigor not only strengthens the validity of our findings but also highlights the importance of using cross-validation as a standard practice in machine learning to achieve optimal performance and reliability.

This reinforces the synergistic effectiveness of integrating linguistic features and employing feature selection techniques, validating their pivotal roles in optimizing the overall performance of the classification process. Fig. 10 presents the result analysis of the proposed framework, which combines linguistic features and feature selection via GA.

D. COMPARATIVE ANALYSIS

In this section, the accuracy of the proposed framework is compared to some recent and existing studies in the literature on the same dataset [4], [5]. Table 18 summarizes the results, demonstrating that the proposed framework performs better than existing studies in classification accuracy and other evaluation metrics. Ji et al. [4] extracted five subsets of features and used six classifiers, with the XGB classifier performing the best with all five feature sets (95.71%). Michael et al. [5] created a machine learning experimental approach based on six baseline methods and three extracted feature sets, as well as a deep learning approach. The deep learning approach with Word2vec achieves 93.8% accuracy.

Upon examining Table 18, the proposed framework outperforms previous studies in terms of detecting suicidal ideation, with an accuracy of 98.92%, as well as results in other evaluation metrics such as accuracy, recall, and f1-score. These studies had some obvious limitations, which are resolved in the proposed framework. For example, in our experiment, GA was used to select the best features,

TABLE 18. Comparison with the previous studies using the same dataset.

Ref	Features Sets	Feature Selection	Best Accuracy	Classifier
[4]	Statistics + TF-IDF + POS + Topics + LIWC	No	95.71%	XGB
[5]	Word2vec	No	93.8%	LSTM-CNN
[5]	Statistics + TF-IDF + BoW	No	88.3%	XGBoost
Proposed	TF-IDF + Topics + N-gram	GA	98.92%	RF

TABLE 19. Comparison with other recent studies.

Ref	Model	Accuracy
Liu et al. [30]	SVM	80.61%
Haque et al. [31]	BiLSTM	93.6%
Chatterjee et al. [32]	LR	87%
Valeriano et al. [35]	LR	79%
Mirtaheri et al. [36]	AL-BTCN	95%
Aldhyani et al. [40]	CNN–BiLSTM	95%
Renjith et al. [41]	LSTM-CNN	90.3%
Proposed	RF	98.92%

in addition to their focus on the level of words rather than the meaning, which is solved by using only the features associated with the meaning.

Table 19 provides a comparison between the proposed method and recent studies on suicide ideation detection. The table demonstrates that the proposed method outperforms the existing studies, achieving higher accuracy and indicating its superior effectiveness in identifying suicidal ideation.

V. CONCLUSION

Mental health concerns, such as depression, anxiety, and ideations of suicide, are becoming increasingly worrisome in modern society. Without proper treatment, severe mental disorders can greatly increase the risk of suicide. Early detection of suicidal thoughts is among the most effective approaches to preventing suicide. As social media continues to evolve, an increasing number of individuals are utilizing online platforms to openly share their emotions and experiences. Social media platforms have provided users with the ability to express themselves freely and anonymously, granting them the option to hide their identity while sharing their thoughts, including suicidal ideations. Anonymous websites and online communities offer a safe space for people to interact with others using asynchronous communication. To detect suicidal ideation, we proposed a new framework to improve the performance of suicidal ideation. It also overcomes the mentioned limitations. It consists of a set of consecutive phases: data preprocessing, feature engineering, and classification. During the data preprocessing phase, we cleaned the row text and prepared it for the feature engineering phase. The feature engineering phase consists of two subphases: (1) feature extraction: various sets of robust features, encompassing statistical, POS, TF-IDF, N-grams, and topic features were extracted. (2) Feature selection: we utilized the GA to select the best and most important features from both the original features set and the linguistic features set. During the classification phase, four classifiers are used: KNN, RF, GBDT, and XGB. We used these classifiers and the different generated feature sets to detect suicidal ideations. The classifiers were trained and tested using these different feature sets. The proposed framework has demonstrated significant effectiveness in detecting suicidal ideation on social media compared to existing studies in the field.

LIMITATIONS AND FUTURE WORK

- **Limitations** There are some limitations to this study as well. Due to Reddit's privacy settings, the inability to know users' age, gender, location, or other information. Another constraint is a shortage of data. One of the most urgent issues in the current study, where supervised learning methods are primarily used, is data scarcity. Manual annotation is usually required. However, there is not enough annotated data to support additional research.
- **Future works** Other factors in detecting suicidal thoughts should be employed in future research. Even though informative feature sets are extracted and a good feature selection method is used, other factors, such as historical data or user posts, temporal information, and spatial information, must be considered. Other methods for selecting features can also be used to enhance the accuracy of suicide ideation detection on online social media. Also, we can propose a hybrid model that combines features from both social media and conventional source data. Initial studies indicate that this integrated approach could improve overall performance by capturing a wider range of indicators.

ACKNOWLEDGMENT

This research was supported by King Saud University Researchers Supporting Project number (RSP2024R444), King Saud University, Riyadh, Saudi Arabia. The authors express their gratitude to King Saud University Researchers Supporting.

REFERENCES

[1] S. Kline and T. Sabri, "The decriminalisation of suicide: A global imperative," *Lancet Psychiatry*, vol. 10, no. 4, pp. 240–242, Apr. 2023.
[2] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2018.
[3] S. T. Rabani, Q. R. Khan, and A. Khanday, "A nove approach to predict the level of suicidal ideation on social networks using machine and ensemble learning," *ICTACT J. SOFT Comput.*, vol. 11, no. 2, pp. 2288–2293, 2021.
[4] S. Ji, C. P. Yu, S.-F. Fung, S. Pan, and G. Long, "Supervised learning for suicidal ideation detection in online user content," *Complexity*, vol. 2018, no. 1, Jan. 2018, Art. no. 6157249.
[5] M. M. Tadesse, H. Lin, B. Xu, and L. Yang, "Detection of suicide ideation in social media forums using deep learning," *Algorithms*, vol. 13, no. 1, p. 7, Dec. 2019.
[6] K. Loveys, P. Crutchley, E. Wyatt, and G. Coppersmith, "Small but mighty: Affective micropatterns for quantifying mental health from social media language," in *Proc. 4th Workshop Comput. Linguistics Clin. Psychol. From Linguistic Signal Clin. Reality*, 2017, pp. 85–95.

[7] F. M. Plaza-del-Arco, M. T. Martín-Valdivia, L. A. Ureña-López, and R. Mitkov, "Improved emotion recognition in Spanish social media through incorporation of lexical knowledge," *Future Gener. Comput. Syst.*, vol. 110, pp. 1000–1008, Sep. 2020.

[8] K.-W. Fu, Q. Cheng, P. W. C. Wong, and P. S. F. Yip, "Responses to a self-presented suicide attempt in social media," *Crisis*, vol. 34, no. 6, pp. 406–412, Nov. 2013.

[9] K. Kang, C. Yoon, and E. Yi Kim, "Identifying depressive users in Twitter using multimodal analysis," in *Proc. Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2016, pp. 231–238.

[10] R. Martínez-Castaño, J. C. Pichel, and D. E. Losada, "A big data platform for real time analysis of signs of depression in social media," *Int. J. Environ. Res. Public Health*, vol. 17, no. 13, p. 4752, Jul. 2020.

[11] X. Huang, L. Xing, J. R. Brubaker, and M. J. Paul, "Exploring timelines of confirmed suicide incidents through social media," in *Proc. IEEE Int. Conf. Healthcare Informat. (ICHI)*, Aug. 2017, pp. 470–477.

[12] M. De Choudhury, E. Kiciman, M. Dredze, G. Coppersmith, and M. Kumar, "Discovering shifts to suicidal ideation from mental health content in social media," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2016, pp. 2098–2110.

[13] M. J. Vioules, B. Moulahi, J. Aze, and S. Bringay, "Detection of suicide-related posts in Twitter data streams," *IBM J. Res. Develop.*, vol. 62, no. 1, pp. 7:1–7:12, Jan. 2018.

[14] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. Eur. Conf. Mach. Learn.*, 1998, pp. 137–142.

[15] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002.

[16] H. Uğuz, "A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm," *Knowl.-Based Syst.*, vol. 24, no. 7, pp. 1024–1032, Oct. 2011.

[17] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining Text Data*. New York, NY, USA: Springer, 2012, pp. 163–222.

[18] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[19] A. K. Uysal and S. Gunal, "A novel probabilistic feature selection method for text classification," *Knowl.-Based Syst.*, vol. 36, pp. 226–235, Dec. 2012.

[20] K. Javed, H. A. Babri, and M. Saeed, "Impact of a metric of association between two variables on performance of filters for binary data," *Neurocomputing*, vol. 143, pp. 248–260, Nov. 2014.

[21] R. F. Kashef, *Cooperative Clustering Model and Its Applications*. Waterloo, ON, Canada: Univ. of Waterloo, 2008.

[22] A. K. Uysal and S. Gunal, "Text classification using genetic algorithm oriented latent semantic features," *Expert Syst. Appl.*, vol. 41, no. 13, pp. 5938–5947, Oct. 2014.

[23] M. Saeed, K. Javed, and H. Atique Babri, "Machine learning using Bernoulli mixture models: Clustering, rule extraction and dimensionality reduction," *Neurocomputing*, vol. 119, pp. 366–374, Nov. 2013.

[24] J. A. Talbott, "Cross-national prevalence and risk factors for suicidal ideation, plans and attempts," *Yearbook Psychiatry Appl. Mental Health*, vol. 2009, p. 186, Jan. 2009.

[25] D. Sikander, M. Arvaneh, F. Amico, G. Healy, T. Ward, D. Kearney, E. Mohedano, J. Fagan, J. Yek, A. F. Smeaton, and J. Brophy, "Predicting risk of suicide using resting state heart rate," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, Dec. 2016, pp. 1–4.

[26] N. Jiang, Y. Wang, L. Sun, Y. Song, and H. Sun, "An ERP study of implicit emotion processing in depressed suicide attempters," in *Proc. 7th Int. Conf. Inf. Technol. Med. Educ. (ITME)*, Huangshan, China, Nov. 2015, pp. 37–40.

[27] R. C. O'Connor and M. K. Nock, "The psychology of suicidal behaviour," *Lancet Psychiatry*, vol. 1, no. 1, pp. 73–85, 2014.

[28] K. D. Varathan and N. Talib, "Suicide detection system based on Twitter," in *Proc. Sci. Inf. Conf.*, Aug. 2014, pp. 785–788.

[29] M. Cabello, M. Miret, J. L. Ayuso-Mateos, F. F. Caballero, S. Chatterji, B. Tobiasz-Adamczyk, J. M. Haro, S. Koskinen, M. Leonardi, and G. Borges, "Cross-national prevalence and factors associated with suicide ideation and attempts in older and young-and-middle age people," *Aging Mental Health*, vol. 24, no. 9, pp. 1533–1542, Sep. 2020.

[30] J. Liu, M. Shi, and H. Jiang, "Detecting suicidal ideation in social media: An ensemble method based on feature fusion," *Int. J. Environ. Res. Public Health*, vol. 19, no. 13, p. 8197, Jul. 2022.

[31] R. Haque, N. Islam, M. Islam, and M. M. Ahsan, "A comparative analysis on suicidal ideation detection using NLP, machine, and deep learning," *Technologies*, vol. 10, no. 3, p. 57, Apr. 2022.

[32] M. Chatterjee, P. Samanta, P. Kumar, and D. Sarkar, "Suicide ideation detection using multiple feature analysis from Twitter data," in *Proc. IEEE Delhi Sect. Conf. (DELCON)*, Feb. 2022, pp. 1–6.

[33] S. T. Rabani, Q. R. Khan, and A. M. U. D. Khanday, "Detection of suicidal ideation on Twitter using machine learning & ensemble approaches," *Baghdad Sci. J.*, vol. 17, no. 4, p. 1328, Dec. 2020.

[34] E. R. Kumar, K. V. S. N. R. Rao, S. R. Nayak, and R. Chandra, "Suicidal ideation prediction in Twitter data using machine learning techniques," *J. Interdiscipl. Math.*, vol. 23, no. 1, pp. 117–125, Jan. 2020.

[35] K. Valeriano, A. Condori-Larico, and J. Sulla-Torres, "Detection of suicidal intent in Spanish language social networks using machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 4, pp. 1–8, 2020.

[36] S. L. Mirtaheri, S. Greco, and R. Shahbazian, "A self-attention TCN-based model for suicidal ideation detection from social media posts," *Expert Syst. Appl.*, vol. 255, Dec. 2024, Art. no. 124855.

[37] P. Boonyarat, D. J. Liew, and Y.-C. Chang, "Leveraging enhanced BERT models for detecting suicidal ideation in Thai social media content amidst COVID-19," *Inf. Process. Manage.*, vol. 61, no. 4, Jul. 2024, Art. no. 103706.

[38] T. Ghosh, M. H. A. Banna, M. J. A. Nahian, M. N. Uddin, M. S. Kaiser, and M. Mahmud, "An attention-based hybrid architecture with explainability for depressive social media text detection in Bangla," *Expert Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 119007.

[39] K. Jawad, R. Mahto, A. Das, S. U. Ahmed, R. M. Aziz, and P. Kumar, "Novel cuckoo search-based metaheuristic approach for deep learning prediction of depression," *Appl. Sci.*, vol. 13, no. 9, p. 5322, Apr. 2023.

[40] T. H. H. Aldhyani, S. N. Alsubari, A. S. Alshebami, H. Alkahtani, and Z. A. T. Ahmed, "Detecting and analyzing suicidal ideation on social media using deep learning and machine learning models," *Int. J. Environ. Res. Public Health*, vol. 19, no. 19, p. 12635, Oct. 2022.

[41] S. Renjith, A. Abraham, S. B. Jyothi, L. Chandran, and J. Thomson, "An ensemble deep learning technique for detecting suicidal ideation from posts in social media platforms," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9564–9575, Nov. 2022.

[42] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.

[43] S. Maruf, K. Javed, and H. A. Babri, "Improving text classification performance with random forests-based feature selection," *Arabian J. Sci. Eng.*, vol. 41, no. 3, pp. 951–964, Mar. 2016.

[44] N. Bidi and Z. Elberrichi, "Feature selection for text classification using genetic algorithms," in *Proc. 8th Int. Conf. Modeling, Identificat. Control (ICMIC)*, Nov. 2016, pp. 806–810.

[45] D. Tufis, and R. Ion, "Part-of-Speech Tagging," in *The Oxford Handbook of Computational Linguistics*. Oxford, U.K.: Oxford University Press, 2022.

[46] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[47] Z. Toosinezhad, M. Mohamadpoor, and H. Tabatabaee Malazi, "Dynamic windowing mechanism to combine sentiment and N-gram analysis in detecting events from social media," *Knowl. Inf. Syst.*, vol. 60, no. 1, pp. 179–196, Jul. 2019.

[48] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Rel. Eng. Syst. Saf.*, vol. 91, no. 9, pp. 992–1007, Sep. 2006.

[49] B. Keshanchi, A. Souri, and N. J. Navimipour, "An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: Formal verification, simulation, and statistical testing," *J. Syst. Softw.*, vol. 124, pp. 1–21, Feb. 2017.

[50] R. L. Haupt, "Optimum population size and mutation rate for a simple real genetic algorithm that optimizes array factors," in *Int. Symp. Transmitting Waves Prog. Next Millennium. Dig. Held Conjunct With USNC/URSI Nat. Radio Sci. Meeting*, vol. 2, Sep. 2000, pp. 1034–1037.

[51] D. Whitley, "A genetic algorithm tutorial," *Statist. Comput.*, vol. 4, no. 2, pp. 65–85, Jun. 1994.

[52] A. S. Ghareb, A. A. Bakar, and A. R. Hamdan, "Hybrid feature selection based on enhanced genetic algorithm for text categorization," *Expert Syst. Appl.*, vol. 49, pp. 31–47, May 2016.

[53] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, 1992.

[54] L. Breiman and R. A. Cutler, "Random forests machine learning," *J. Clinical Microbiol.*, vol. 2, pp. 199–228, Oct. 2001.
[55] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *Proc. OTM Confederated Int. Conf., Move Meaningful Internet Syst.*, Catania, Italy, Nov. 2003, pp. 986–996.
[56] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.
[57] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
[58] S. Ji, S. Pan, X. Li, E. Cambria, G. Long, and Z. Huang, "Suicidal ideation detection: A review of machine learning methods and applications," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 1, pp. 214–226, Feb. 2021.
[59] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, Dec. 2020.

ABDALLAH BASYOUNI received the B.Sc. degree (Hons.) in information systems from the Faculty of Computers and Information, Menofia University, Menofia, Egypt, in 2016. He is currently an Assistant Lecturer with the Faculty of Computers and Information, Menoufia University. His research interests include suicide detection, natural language processing, machine learning, and deep learning.

HATEM ABDULKADER received the B.Sc. and M.Sc. (by Research) degrees in electrical engineering from the Faculty of Engineering, Alexandria University, Egypt, in 1990 and 1995, respectively, and the Ph.D. degree in electrical engineering from the Faculty of Engineering, Alexandria University, in 2001, specializing in neural networks and applications. He is currently a Professor with thee Information Systems Department, Faculty of Computers and Information, Menoufia University, since 2004. He has worked on a number of organizations. He has contributed more than 150 technical articles in the areas of neural networks, big data, database applications, information security, and internet applications.

WAIL S. ELKILANI received the M.Sc. and Ph.D. degrees from the Faculty of Engineering, Cairo University, Egypt, in 1996 and 2001, respectively. He has held a postdoctoral research position with the Advanced Networking Laboratory, The University of British Columbia, Vancouver, Canada, in 2006. Since 2015, he has been a Full Professor with the College of Applied Computer Science, King Saud University, Riyadh, Saudi Arabia. He has supervised nearly 20 master's and Ph.D. students. He has authored over 30 articles on networking and computer systems in international journals and conferences. His research interests include ad-hoc wireless networks, internet protocols, and systems performance evaluation of parallel and networked computer systems. Lately, he is working in exploring hacking threats in wired LANs, WLANs, and VoIP networks. He was chosen as a member of the Cisco Advisory Board, in 2009 and 2010. He was awarded by Cisco Systems as the Best Instructor of Cisco Academies, in 2006 and 2008, for his innovation in integrating Cisco curriculums with academic networking courses.

ABDULLAH ALHARBI received the Master of Science degree in information technology from Rochester Institute of Technology, Rochester, NY, USA, and the master's degree in information assurance and cybersecurity and the Ph.D. degree in computer science from Florida Institute of Technology, Melbourne, FL, USA. He is currently an Associate Professor in computer science with King Saud University, Riyadh, Saudi Arabia. He is also the Dean of the College of Applied Computer Sciences, King Saud University (Muzahmiyah Branch). He is also the CEO of the Information Security Association (Hemaya), a non-profit organization, Saudi Arabia. He is also a Research Fellow with the Center of Excellence for Information Assurance, King Saud University, where he was the Department of Administrative Sciences Chair, Community College. His research interests include wearable devices security, transparent and continuous security, alternative authentication, usable security, and behavioral biometrics. He also received the Information Assurance and Cybersecurity Graduate Certificate for the master's degree.

YULONG XIAO received the bachelor's degree from Minjiang University, in 2022. He is currently pursuing the master's degree in communication engineering with Chongqing University of Posts and Telecommunications. His research interests include wireless communication and channel modeling.

ASMAA H. ALI received the B.Sc. degree from the Faculty of Computers and Information, Helwan University, Egypt, in 2000, the M.Sc. (by Research) degree in computers and information from the Faculty of Computers and Information, Menoufia University, Egypt, in 2007, specialized in intelligent systems and its applications, and the Ph.D. degree in computers and information from the Faculty of Computers and Information, Menoufia University, in 2017, specialized in software engineering. She has been a Lecturer with the Information Systems Department, Faculty of Computers and Information, Menoufia University, since 2017. Her research interests include decision support systems and expert systems, data mining, database design and management, software engineering, blockchain, system analysis and design, data integration, information retrieval, project management, and business intelligence. She has been a Reviewer of *IET Software* journal and *The Open Biomedical Engineering Journal* (IET), since 2018.