



An evolutionary approach for depression detection from Twitter big data using a novel deep learning model with attention based feature learning mechanism

Prabhakar K^a and Kavitha V^b

^aDept. of CSE, School of Engineering and Technology, CMR University, Bangalore, India; ^bDepartment of Computer Science and Engineering, University College of Engineering Kancheepuram, Kancheepuram, India

ABSTRACT

One of the main factors causing suicide is depression. However, many cases of depression go undiagnosed because they are not correctly diagnosed. An increasing number of people with mental illnesses express their emotions online using tools like social media (SM) and specialized websites. Recently, efforts have been made to use Machine Learning (ML) and deep learning (DL) models to predict depression from SM platforms. However, it is problematic that most ML algorithms now provide no explanation. As a result, this study proposes a novel Deep Learning (DL) model called residual network 50, which includes optimal long short-term memory (RNT-OLSTM) for Depression Detection (DD) on Twitter data. In addition, to address the issue of data imbalance in the Twitter data, a cluster-based oversampling approach is used, which considerably reduces the possibility of bias towards the dominant class (non-depressed).. Finally, the embedding layers are inputted to RNT-OLSTM for DD, in which the hyperparameters of the network are tuned using the Sine Chaotic map and constriction factor-based Coyote Optimization Algorithm (SCCOA) to minimize the prediction loss. The out-comes prove that the proposed system performs better than the existing schemes for the DD of imbalanced Twitter data with higher detection rates.

ARTICLE HISTORY

Received 24 September 2023
Accepted 10 December 2023

KEYWORDS

Depression detection;
Twitter; social media; big
data; deep learning; machine
learning; data balancing; and
word embedding

1. Introduction

Users have altered their methods of communication and information gathering on the web due to the rapid development of computer science and Internet technologies. Through blogs and SM, consumers now actively participate in making digital content in the Web 2.0 Era. Utilizing the features offered by these new platforms, such as Facebook, Twitter, Reddit or related sites [1–3], users post their thoughts, ideas and sentiments about a subject, topic or issue online [4]. SM content from different websites and platform providers, such as posts, comments, posts, tweets and reviews, has dramatically influenced the development of big data. Data analytics and artificial intelligence are currently dealing with a new era of excitement due to the explosion of big data from SM. Finding effective methods to gather a high volume of social data and draw insights from the enormous amount of data obtained is the main problem faced by big social data.

By capturing public opinions, DD in big social data might offer commercial insights. *Depression* is a severe mental disorder that has a detrimental effect on one's feelings and thoughts [5,6]. In 2021, the World Health Organization predicted that 280 million individuals worldwide would experience depression. Among them went untreated in nations with low and moderate

incomes. During the coronavirus epidemic, the number of instances of depression in the US has tripled. In Thailand, 60% of suicides are motivated by depression, and the number of cases of depression is still rising [7].

Due to a lack of awareness of depression, there are still numerous occurrences of misdiagnosed depression, even though many patients are receiving treatment. As a result, patients will experience negative thoughts, including self-isolation, erratic behaviour and suicidal thoughts and will become dependent on drugs like antidepressants [8,9]. Predicting is crucial before becoming impacted by a psychological disorder [10]. Data cleansing techniques are more appropriate for detecting depression. The majority of label-cleaning techniques, however, rely on or make assumptions about the dataset's noise distribution. As a result, numerous researchers have recently focused ML algorithms on diagnosing depression [11,12]. ML (ML) is one of the most important aspects of artificial intelligence [13]. Finding the most helpful term features to make a forecast has demonstrated exemplary performance in predicting whether SM people have depression [14].

In order to do so, a variety of ML approaches, such as random forest (RF), logistic regression (LR), naive-Bayes (NB) and support vector machines (SVM) [15],

can be employed to identify depression. The effectiveness of these algorithms can be assessed using a confusion matrix. A successful algorithm will have a high precision score and help accurately forecast sentiment, which can be either positive or negative [16]. However, it relies on the grammatical accuracy of SM data and requires many attributes to diagnose depression. It also only recognizes explicit aspects while failing to extract implicit ones. Several writers have recently focused on DL, an area of ML, for the early DD in SM data [17,18]. Even without feature extraction, DL models can handle high-dimensional data with little information loss [19].

To address the classification problem, DL models do not require the described classifiers of hand-modelled models [20]. It was created to accurately forecast depression from SM data by training a non-linear, complex connection among input factors utilizing their prior features. This strategy needed a more sophisticated model to study the effects of various parameters and their modifications. The issue of class imbalance is also a significant task in detecting depression. Still, the focused class imbalance problem is only partially resolved because the imbalanced data hinders the classifier's performance, which also lowers the prediction rate. This research subsequently suggests a novel DL method for DD of Twitter data. The proposed work's primary contributions are listed below.

- The SDFCM-SMOTE technique is developed to lower the computational overhead of the training model and tackle the class imbalance problem in the gathered Twitter dataset.
- A word embedding approach based on TF-IDF and W2V is provided to express user comments as a piece of text in a continuous vector space.
- To improve classification accuracy, develop an effective feature extraction approach based on ResNet50 for extracting features from word-embedded vectors.
- To propose an OLSTM network for predicting depressed symptoms in SM patients. SCCOA, which tunes the network's hyperparameters, is proposed to reduce prediction loss in depression diagnosis.

The following outlines the remaining parts of the paper: Section 2 surveys the recently developed ML and DL models to perform DD of SM data. A brief explanation of the suggested research model is given in section 3. The outcomes of the proposed and existing research frameworks regarding some performance metrics are given in section 4. At last, the conclusion and future research directions are described in section 5.

2. Related works

This section discusses the recent methodologies of DD with the use of various ML and DL models. The limitations of the surveyed models are presented at the end of the literature, with a solution offered by the proposed system to overcome those limitations.

Lei Tong et al. [21] presented a DD approach called cost-sensitive boosting pruning trees (CBPT) for Twitter SM data. Initially, the system performed pre-processing operations such as reducing influencing noisy samples, stemming, and removing irregular words. Afterwards, the features such as the user profile and linguistic and social interaction were extracted from the pre-processed data. Finally, the extracted features were fed into the CBPT for DD. The system utilized two datasets, as CLPsych 2015 (CLPsych2015) Twitter Dataset and Tsinghua Twitter Depression Dataset (TTDD) from the UCI repository. The outcomes illustrated that the method attained a higher accuracy and f-score than the existing related schemes.

Amna Amanat et al. suggested a DL strategy for DD from textual data [22]. The data was initially gathered from the freely available Tweets-Scraped dataset on Kaggle, which included more than 4000 tweets. After that, mentions, stop words, retweets, and URLs were deleted from the dataset as a portion of pre-processing. Then, features from the dataset were extracted using the one-hot encoding. Then feature visualization was done using principal component analysis (PCA). Finally, a recurrent neural network (RNN) and long short-term memory (LSTM) were used for DD. The outcomes showed that implementing the system with a lower false positive rate attained an accuracy of 99%.

Based on the multimodal analysis, RaminSafa et al. [23] demonstrated an automatic detection of depression symptoms on Twitter. Data was initially gathered from tweets using a regular expression on a self-reported diagnosis. All retweets and URLs were eliminated, and the sentiment polarity score was then determined using a lexicon-based methodology. The correlation characteristic was then measured using harmonic analysis, and the dimension was reduced by the singular value decomposition (SVD) method. Finally, a multimodal ML technique was used to perform the DD. The method attained an accuracy of 91% and 83% when worked independently on bio-text and tweets, which were satisfactory compared to the results of the existing approaches.

To detect depression, Raymond Chiong et al. [24] investigated sentiment lexicons and content-based characteristics for DD. The input features were extracted initially from the collected data. All input features were subsequently normalized using min-max normalization on a scale of 0–1. In the end least, four

standard single classifiers such as LR, decision tree (DT), multilayer perceptron (MLP) and SVM, as well as four ensemble models say adaptive boosting (AB), bagging predictors (BP), gradient boosting (GB) and RF were used to find depression in tweets. Experiments were conducted on two openly available Twitter datasets, and findings showed that the proposed system attained an accuracy of 96%, which was superior to the existing detection techniques.

Hamad Zogan et al. [25] presented an explainable multi-aspect DD with a hierarchical attention network (MDHAN) for the DD of SM data. The system first gathered three complementary datasets, including depression, non-depression, and candidate datasets for depression. After that, the dataset's social data and interactions, Emoji sentiment, topic distribution, and domain-specific features were retrieved. The user tweets encoder was then carried out using RNN. The RNN's classification layer assessed whether the user was depressed or not. The experimental results demonstrated that the MDHAN outperformed previous approaches with 0.895 accuracies, 0.902 precision, 0.892 recall, and 0.893 f-measure, ensuring substantial evidence to support the prediction.

Multimodal DD on Instagram was suggested by Chun Yueh Chiu et al. [26], considering the period between the posts. The system gathered a dataset of depressive and non-depressive Instagram users to make DD on SM. Then, data pre-processing operations such as word segmentation, redundancy removal, and emoticon decoding were carried out. The pre-processed dataset was then used to extract features via the convolutional neural network (CNN). Finally, bidirectional LSTM was employed to identify the person with depression. The system may detect depressive users with an F1 score of up to 0.835, according to experimental results.

Raymond Chiong et al. [27] suggested an ML model incorporating a textual featuring learning strategy for DD. The technique was first used to remove punctuation, numerals, stop words, and word correction. The machine then utilized the BOW approach to extract features from the dataset. Finally, DD was carried out using logistic regression, SVM, MLP, decision tree, RF, adaptive boosting, bagging, and gradient boosting approaches. Initially, two openly available Twitter datasets were utilized for training and testing the ML classifiers. After that, the system used three non-twitter SM data sources such as Reddit, Facebook and an electronic diary to validate the presented classification approaches. According to test results, the system could accurately and successfully detect depression in SM texts or data in texts with an accuracy of 99.28%.

A summarization-boosted deep framework for DD on SM was introduced by Hamad Zogan et al. [28].

The dataset's most pertinent embedded summary sentence features were first extracted using CNN. The Bidirectional Gated Recurrent Units (BiGRU), an RNN, was then utilized to extract the long-term dependencies and sequential information from the features extracted using the CNN. After that, the dimension was decreased using the Latent Dirichlet Allocation (LDA). Finally, the DD was done using the fully connected layer of the BiGRU. A dataset of depression, non-depression, and depression candidates was used in the investigation. The system did comprehensive experiments, and the results showed that it performed substantially better than existing methods (+3% accuracy and +6.5% F-Score).

A DD system based on ML and natural language processing over social interactions was provided by Tanya Nijhawan et al. [29]. The system first performed tokenization, the elimination of undesirable patterns, the elimination of stop words, and stemming. Then, using the Bag-of-words method, the system extracted feature vectors. Finally, using ML models and Bidirectional Encoder Representations from Transformers (BERT), the depressed users were detected. Some tweets made up the dataset used to train the binary sentiment analysis system. The experimental findings demonstrated the system's accuracy of 97.78%, which was higher than the state-of-the-art techniques.

Using the naive multinomial theorem, Rinki Chatterjee et al. [30] suggested DD from SM posts. The system first retrieved data based on the Facebook API and the Twitter API. The system counted how many positive, negative, and neutral feelings there were overall. Following the classification of the remarks, the level of depression and non-depression was discovered. Finally, the system detected the depression using the naive multinomial theorem. The experimental findings demonstrated that the system outperformed traditional approaches regarding precision, f-measure accuracy, and recall achieving values of 0.77, 0.86, 0.31, and 0.46.

The ML and DL-based technique for DD has been the focus of recent publications. Numerous authors have suggested various ML techniques for diagnosing depression, including decision trees, NB, SVM, K-nearest neighbours, and LR. These ML-based methods produce higher outcomes, but they struggle to identify depressed individuals at the user level, making them vulnerable to inaccurate prediction [21–28]. Handling the enormous amount of data provided by Twitter is a difficult challenge. Most authors use DL-based techniques for depression identification to get around these problems [29–33]. The distribution might be increased to obtain great accuracy. As a result, the DL approaches are more reliable and offer great precision for severe depression utilizing human history. However, none of those mentioned above studies focused on tweaking

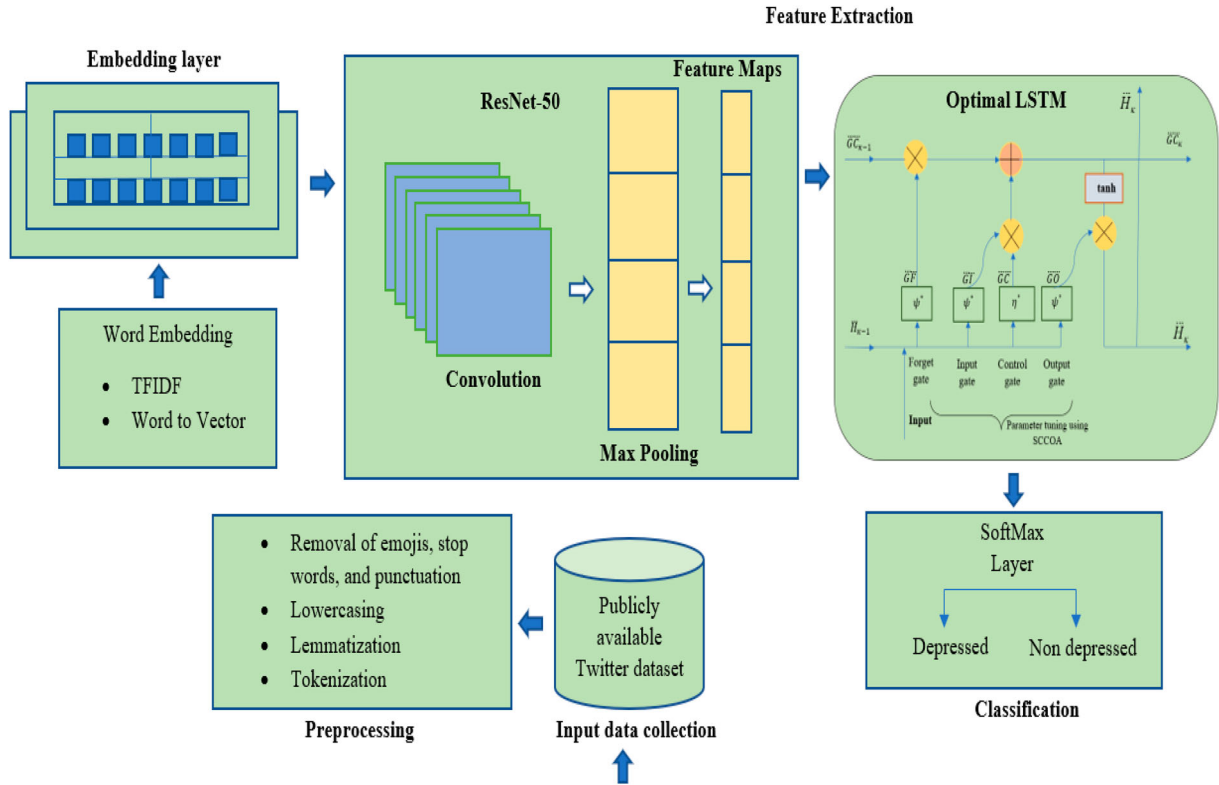


Figure 1. Structural design of the proposed system.

the random initialization hyperparameter of the DL model, which affects the system's performance. As a result, the approaches mentioned above that are advised produce predictions with less accuracy and need more processing time. The suggested method uses a novel DL technique, called OLSTM, to address these flaws, and SCCOA is used to modify the LSTM's hyperparameter. Additionally, the low prevalence of depression in the general population and the resulting uneven class composition present challenging issues for the DL techniques. Previous research has yet to look into the issue of class imbalance. An unbalanced dataset can reduce the model's accuracy. With the help of the SDFCM-SMOTE technique, our suggested study resolves this problem.

3. Proposed methodology

This paper proposes a novel deep-learning model for the DD of SM data. Initially, the Twitter data is collected from the publicly available Kaggle dataset. Afterwards, the class imbalance problem is solved by SDFCM-SMOTE. Then, the pre-processing operations, such as removing emojis, stopping words and punctuation, lowercasing, lemmatization, and tokenization, were performed on the balanced dataset. Then word embedding is performed on the pre-processed data utilizing TF-IDF and W2V, which gives the feature representation of pre-processed data. Finally, the RNT-OLSTM is employed for DD, where, ResNet50 is utilized to extract spatial features, and SCCOA tunes the hyperparameter

of LSTM. The structure diagram of the proposed system is shown in Figure 1.

3.1. Dataset balancing

Initially, Twitter data is gathered from publicly available data sources. The class imbalance problem of the dataset is rectified using the proposed clustering model. A class-imbalanced dataset is one in which the majority class surpasses the minority class. This severe issue can lead to overfitting problems and a high false positive rate (FPR) in the classification task. This article suggests a SDFCM-SMOTE to solve the class imbalance issues. The suggested algorithm includes three steps: clustering, filtering, and oversampling. In the first step, clustering, SDFCM is used to build clusters using the acquired Twitter data as input. One of the most well-known algorithms, FCM, performs clustering effectively by minimizing the objective function and iterating membership and centroid. It builds a k-numbers of clusters before assigning each data point to a cluster. The choice of the first cluster centre and the beginning membership value affects how well the FCM algorithm performs. The FCM algorithm will converge very quickly, and processing time can be greatly decreased if an initially chosen cluster centre is near the original final cluster centre.

The clustering results are highly unpredictable because the initial cluster centres are typically chosen at random by the conventional FCM algorithm. So, the optimal k-value is first determined using the

Silhouette Distance before FCM clustering is applied. These silhouette distance-based FCMs for selecting optimal cluster centroids are named SDFCMs. The proposed algorithm's next phase involves filtering procedures that keep clusters with a high percentage of minority samples and synthesize more minority samples into sparse clusters. Finally, the cluster with a low density of minority samples is oversampled using the SMOTE method. These steps are thoroughly discussed in the section below.

Step 1: Clustering

Let us take the input Twitter dataset as $\overline{IT}_s'' = \{x_1, x_2, x_3, \dots, x_A\}$ and then the FCM algorithm divides the samples into creates a k - number of random clusters. However, this randomly creates initial clusters that likely get stuck in a local optimum and never converge to the global optimum. Hence the proposed system used silhouette distance to create an optimal number of cluster centres. The SD's value ranges from 1 to +1, where +1 denotes the best fit for the cluster, and -1 denotes the worst fit and which is expressed as:

$$SD = \left(\frac{\ddot{u} - \ddot{v}}{\max(\ddot{u}, \ddot{v})} \right) \quad (1)$$

Where, \ddot{u} refers to the average distance between the instances within-cluster and \ddot{v} refers to the average distance between the instance and the nearest cluster or clusters. The FCM algorithm's objective function is therefore defined as follows:

$$\min \overline{O}f_{FCM}(x, y) = \sum_{m=1}^A \sum_{n=1}^{C_{om}} x_{mn}^{w_m''} (\ddot{D}_{mn}^2), \quad w_m'' > 1 \quad (2)$$

$$\ddot{D}_{mn}^2 = \|(\overline{IT}_s'')_m - \bar{y}_n\|^2 \quad (3)$$

Where, w_m'' refers to the fuzzy weighted index or smoothing factor. It controls the degree of fuzziness between fuzzy classes, \ddot{D}_{mn}^2 indicates the squared difference between the element $(\overline{IT}_s'')_m$ in the dataset \overline{IT}_s'' and the cluster centre \bar{y}_n . Afterwards, the degree of membership (x_{mn}) can be obtained by minimizing the objective function. Like the pixel probability in a mixed modelling assumption, the degree of membership in fuzzy clustering is crucial. The iterative functions of membership and centroid are given in the following equations.

$$x_{mn} = \left[\sum_{r=1}^A \left(\frac{\ddot{D}((\overline{IT}_s'')_m, \bar{y}_n)}{\ddot{D}((\overline{IT}_s'')_m, \bar{y}_r)} \right)^{2/w_m''-1} \right]^{-1} \quad (4)$$

$$\bar{y}_n = \frac{\sum_{n=1}^{C_{om}} x_{mn}^{w_m''} (\overline{IT}_s'')_m}{\sum_{n=1}^{C_{om}} x_{mn}^{w_m''}} \quad (5)$$

Each sample point's final membership matrix to all class centres is obtained by optimizing the objective function to minimize it. Thus, the best classification for all sample data was found.

Step 2: Filtering

Here, the clusters having a higher number of minority samples are filtered. Then Euclidean distance matrix is computed for each filtered cluster \tilde{C}_α by ignoring the majority data samples. Finally, each cluster's mean distance $d(\tilde{C}_\alpha)$ is computed by summing the distance matrix" of all diagonal elements and then dividing it by the total count of non-diagonal elements. The density of each filtered cluster is computed using equation (6).

$$Density(\tilde{C}_\alpha) = \frac{\overline{MS}(\tilde{C}_\alpha)}{d(\tilde{C}_\alpha)^l} \quad (6)$$

Where, $\overline{MS}(\tilde{C}_\alpha)$ indicates the count of minority samples in the cluster and l - refers to the count of features. Then calculate the sparsity of each filtered cluster using the following equation.

$$Sparsity(\tilde{C}_\alpha) = \left(\frac{1}{Density(\tilde{C}_\alpha)} \right) \quad (7)$$

At last, using equation (8), calculate the weight of each filtered cluster (WF_c).

$$WF_c(\tilde{C}_\alpha) = \frac{Sparsity(\tilde{C}_\alpha)}{\sum_{\alpha=1}^k Sparsity(\tilde{C}_\alpha)} \quad (8)$$

Step 3: Oversampling

Finally, the SMOTE oversampling is performed for each filtered cluster. SMOTE is an oversampling method in which false samples are created for the minority class. This strategy seeks to solve the overfitting problem induced by random oversampling. It concentrates on the feature space to develop new instances by using interpolation between positive instances that are close together. It is mathematically written as follows:

$$WF_c = \hat{p}_m + WF_c(\tilde{C}_\alpha) \times (\hat{q}_m - \hat{p}_m) \quad (9)$$

Where, WF_c indicates a newly generated sample, \hat{p}_m denotes a randomly selected minority sample in the cluster; \hat{q}_m refers to the nearest neighbour minority sample of \hat{p}_m .

3.2. Data pre-processing

Tweets, in particular, frequently include noise and unhelpful information like typos, slang words, user-generated abbreviations and white spaces, which are unwanted and make the learning process harder for DD. To solve these issues, in this paper, the removal of stop words, emojis, punctuation, lowercasing, lemmatization, and tokenization is performed on the balanced dataset, which makes the learning process easier and improves prediction accuracy. These are explained deeply in the following steps:

Step 1: Removal of Emojis, stop word and Punctuation

One of the most often utilized pre-processing stages is stop word removal. Stop words such as “is,” “was,” “at,” “if,” and similar words, which do not contribute any meaning to the sentences, so they are removed from the tweets. It does not only reduce the vector space but also improves the performance by increasing the execution speed, calculations, and also the accuracy. Also, the collected Twitter data may contain emojis. A brief digital image or icon called an emoji can convey a thought or emotion. It must also be removed to improve classification accuracy. The punctuations in the Twitter data have been removed as an additional pre-processing step that treats every text in the tweets equally. Additionally, one more typical pre-processing method helps to treat each sentence similarly by removing punctuation from the Twitter data. After the punctuation has been removed, the words data and data! For instance, they are regarded similarly.

Step 2: Lowercasing

Lowercasing is another pre-processing approach used in natural language processing. It changes the input data into lowercase so that it changes “DATA,” “Data,” “DaTa,” and “DATa” into “data.” Each word in our dataset is first converted to its original general form before being turned into lowercase to maintain consistency and prevent confusion when conducting a DD study.

Step 3: Lemmatization

After that, a lemmatization process is carried out, which converts the word to its meaningful base form to make the system accurate.

Step 4: Tokenization

Finally, tokenization is performed on the dataset to increase the system’s efficiency. Tokenization separates each sentence into parts, such as words, phrases, and other elements.

3.3. Word embeddings

The process of converting words into vectors is known as word embedding. It is a helpful illustration of words and often leads to improving performance in the DD

task. They maintain valuable syntactic and semantic features of all the words. In this work, the proposed system uses TF-IDF and W2V to extract vector representations of words.

3.3.1. TF-IDF

The TF-IDF technique efficiently recovers keywords and determines each token’s importance and relevance in a particular context. It determines the relative importance of each word used in the text. Inverse Text Frequency and Word Frequency are the two mathematical techniques TF-IDF uses. This paper uses TF-IDF because of its superior performance, particularly in increasing accuracy and reducing error values. Set $\overline{\overline{DP}}_n$ points to a set of documents, and $\overline{\overline{dp}}$ denotes a single document ($\overline{\overline{dp}} \in \overline{\overline{DP}}_n$). Every document is represented as a collection of sentences and words $\overline{\overline{WoR}}$. Its first component, TF, analyzes the occurrence of specific words to determine their similarity using equation (10).

$$TF(\overline{\overline{WoR}})_{\overline{\overline{dp}}} = \frac{N_{\overline{\overline{WoR}}}(\overline{\overline{dp}})}{|\overline{\overline{dp}}|} \quad (10)$$

Where, $N_{\overline{\overline{WoR}}}(\overline{\overline{dp}})$ indicates the count of recurrent words $\overline{\overline{WoR}}$ in the document $\overline{\overline{dp}}$ and $|\overline{\overline{dp}}|$ refers to the size of the document $\overline{\overline{dp}}$ and it is computed by using the following equation:

$$|\overline{\overline{dp}}| = \sum_{\overline{\overline{WoR}} \in \overline{\overline{dp}}} N_{\overline{\overline{WoR}}}(\overline{\overline{dp}}) \quad (11)$$

After that, its second component, IDF, which determines how many texts in a corpus includes a particular word, is expressed as follows:

$$IDF(\overline{\overline{WoR}})_{\overline{\overline{dp}}} = 1 + \log \left(\frac{|\overline{\overline{DP}}_n|}{|\{\overline{\overline{dp}} : \overline{\overline{DP}}_n | \overline{\overline{WoR}} \in \overline{\overline{dp}}\}|} \right) \quad (12)$$

The calculation of TF-IDF, which comes last, involves multiplying the TF values by the IDF calculation results for each word. This is mathematically shown as follows:

$$\overrightarrow{WM}_{t,\overline{\overline{dp}}} = TF(\overline{\overline{WoR}})_{\overline{\overline{dp}}} \times IDF(\overline{\overline{WoR}})_{\overline{\overline{dp}}} \quad (13)$$

Where, $\overrightarrow{WM}_{t,\overline{\overline{dp}}}$ indicates the weight of the term (t) in a document $\overline{\overline{dp}}$. This weighting equation is applied to each word in a document.

3.3.2. Word2vec

Another technique for obtaining word embeddings is W2V, which constantly represents words as vectors. This combines efficient use of the continuous

bag-of-words (CBOW) architecture and the skip-gram design for computing word vector representations. The CBOW technique uses a context as input and seeks to forecast a specific word, whereas the skip-gram model uses a word as input and seeks to predict a target context. The similarity between Skip-gram and CBOW is based on the presumption that words with comparable vector values will appear more frequently. Because the window contains word location information, W2V learns word embedding using nearby words.

3.4. Depression detection

Once word embeddings are obtained, DD is performed in this phase. In this paper, the proposed system uses an RNT-OLSTM for DD, which classifies the word-embedded data into depressed and non-depressed tweets. The proposed approach involves “5” layers: the convolution layer, embedding layer, LSTM layer, pooling layer, and SoftMax layer. Here, the Resnet50 extracts feature from the model primarily via convolution and pooling. A CNN with 50 layers is called ResNet-50. By introducing a shortcut to achieve residual learning, the residual block in ResNet50 integrates the features of the input layer with those of the neighbouring output layer. The OLSTM layer then uses the spatial information obtained from Resnet50 to identify the depression. Because LSTM has feedback connections, it can process all of the data in a sequence. However, the choice of hyperparameter of the LSTM model has a significant impact on its performance. Random hyperparameter initialization increases the processing speed and computational complexity. Thus, the proposed system uses the SCCOA approach to optimally select the LSTM model’s hyperparameter. This improvisation in conventional LSTM makes the system better. This combination of ResNet50 and OLSTM, termed RNT-OLSTM, achieved higher classification performance to predict depressed users on Twitter. Figure 2(a) depicted that process of big-data analytic on twitter network for real-time depression and detection.

(a) Embedding Layer

An embedding layer is an example of a hidden layer in a neural network. Said this layer converts input data from a high-dimensional to a lower-dimensional space, enabling the network to better understand the relationships between inputs and process the data. This layer of the model transforms each list associated with a particular word in the information array into an element vector. The embedding layer’s primary responsibility is to generate the following input embedding matrix for each word chosen from the training set:

$$\overline{EL}(\overline{WoR}) = \Re^{V_{S_v} \times V_{D_D}} \quad (14)$$

Where, $\overline{EL}(\overline{WoR})$ indicates the embedding matrix of the word \overline{WoR} , \Re denotes the real number system, V_{S_v} and V_{D_D} refers to the vocabulary size and the dimension of the word embedding vector.

(a) Convolutional Layer

After getting the word embedding matrix, it is fed into the ResNet50 model’s convolutional layer. An entire neural network’s building blocks are created using this layer. The convolution layer’s primary function is to remove spatial characteristics from the word embedding matrix. It performs the initial convolution with 7×7 kernel sizes.

(a) Max Pooling Layer

The max pooling layer is the next. A “maximum pooling” operation finds the highest value in each patch of each feature map. This layer helps the model change the data only with the needed information. Additionally, it aids in reducing overfitting issues that arise in neural networks. It performs the pooling operation with a kernel size of 3×3 .

(a) LSTM layer

LSTM is a kind of recurrent neural network (RNN) which can extract features from a previous layer and use those features to detect depression. The LSTM layer’s four gating layers maintain the cell state, consisting of two input gates, a forget gate, and output gates. The input gate determines which input should be added to the cell state. The forget gate chooses which previous cell state to forget based on the current cell state. It is much less susceptible to the vanishing gradient problem and is very efficient at modeling complex sequential data. Each layer comprises the weights and biases between the input and the hidden layer and between the hidden layer and the output layer. These weights and biases are randomly assigned during the training process. However, these random weights and bias values made the system poor and required more computational power. So, the proposed system uses SCCOA to optimally select the weight and bias of the conventional LSTM model. This optimal selection of the hyperparameter in LSTM is termed OLSTM. These are explained in the following steps. The structure of OLSTM is shown in Figure 2(b).

Step 1: Hyperparameter tuning using SCCOA

The COA is a bio-inspired optimization algorithm that considers the social structure and environmental adaption of coyotes to tackle continuous optimization problems. It is based on the intelligent behaviour of coyotes. However, the standard COA has a slow convergence rate, low solution accuracy, and a tendency

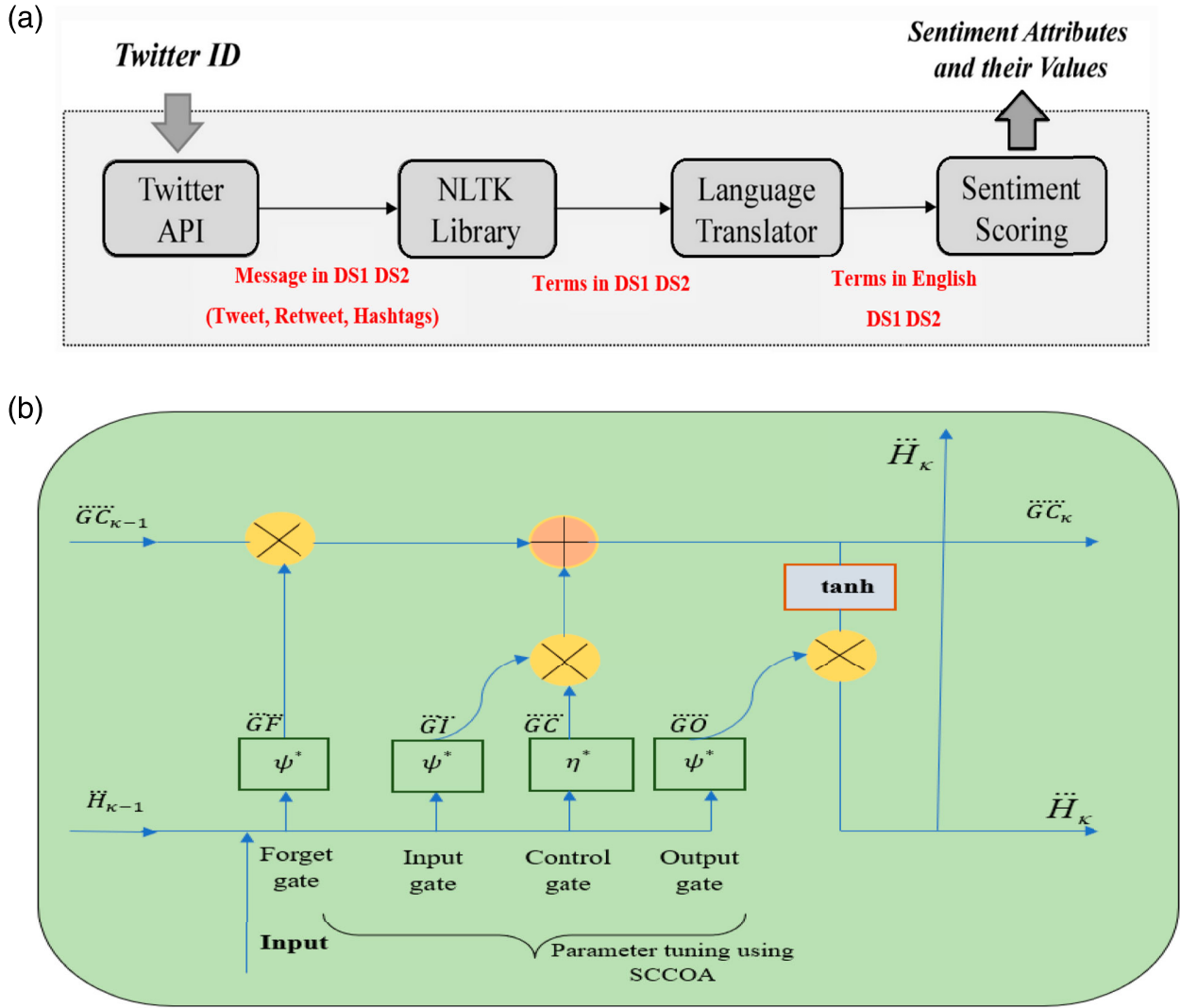


Figure 2. (a): Process of big-data analytic on twitter network for real-time depression. (b): Structure of OLSTM.

to settle on local optimal solutions. So here, a sine chaotic map is utilized to initialize the population of the individuals in the optimal search space that boosts the population diversity of the algorithm. Additionally, the position of the individuals is controlled using the constriction factor to prevent the algorithm from forming an optimal local problem. These two improvements in conventional COA are termed SCCOA. The COA algorithm initializes the population of coyotes using sine chaotic map. It is another type of chaotic map, but the map has problems such as small parameter range and uneven sequence distribution, and it is expressed by,

$$\ddot{R}_a^{e,\xi+1} = f_{\sin e}(\ddot{R}_a^{e,\xi}, \phi) = \sin^2 \left(\beta \arcsin \sqrt{\ddot{R}_a^{e,\xi}} \right) \quad (15)$$

Where, $\ddot{R}_a^{e,\xi+1}$ indicates the position of the a -th coyote in e -th pack at the ξ -th iteration, ϕ and β are considered as 10 and 0.5. Then, compute the fitness of the individuals in the population using equation (16).

$$FitN_a^{e,\xi} = f_{-f}(\ddot{R}_{a,b}^{e,\xi}) \quad (16)$$

Where, f_{-f} indicates the fitness evaluation function. Then, the probability of coyotes leaving or being evicted from the population can be formulated as follows:

$$\hat{\rho}_{pr} = 0.005 \cdot K_a^2 \quad (17)$$

Where, K_a indicates the number of coyotes, where, $K_a \leq \sqrt{200}$ and $\hat{\rho}_{pr}$ are greater than 1. The best coyote at each iteration is considered the alpha coyote ($\ddot{\alpha}$) and that is computed using the equation (18).

$$\ddot{\alpha}^{e,\xi} = \ddot{R}_a^{e,\xi} \text{ for } \min FitN_a^{e,\xi} \quad (18)$$

After that, the cultural tendency ($\overline{CT}_b^{e,\xi}$) of the coyotes is estimated using its median social conditions, which is mathematically expressed using equation (19).

$$\overline{CT}_b^{e,\xi} = \begin{cases} \ddot{M}_{K_a+\frac{1}{2},b}^{e,\xi}, & K_a \text{ is odd} \\ \frac{\ddot{M}_{\frac{K_a}{2},b}^{e,\xi} + \ddot{M}_{\frac{K_a}{2}+1,b}^{e,\xi}}{2}, & \text{otherwise} \end{cases} \quad (19)$$

Where, $\ddot{M}^{e,\xi}$ refers to the coyote's ranked social conditions. Next, a new coyote's birth is estimated as

$$\ddot{N}\ddot{B}_b^{e,\xi} = \begin{cases} \ddot{R}_{\delta_1,b}^{e,\xi} & RaN_b < SP_s \\ \ddot{R}_{\delta_2,b}^{e,\xi} & RaN_b \geq SP_s + AP_p \\ \ddot{D}_b & Otherwise \end{cases} \quad (20)$$

$$\begin{aligned} SP_s &= 1/DiM \\ AP_p &= (1 - SP_s)/2 \end{aligned} \quad (21)$$

Where, SP_s and AP_p refers to the scatter and association probability, δ_1 and δ_2 represents the arbitrary coyotes of the e^{-th} pack, and \ddot{D}_b and RaN denotes the random variables ranges from $[0, 1]$. Each a^{-th} coyote in the e^{-th} pack updates its social condition at each iteration using the following equation (22).

$$\ddot{R}_a^{e,\xi+1} = \begin{cases} \ddot{R}_a^{e,\xi} + \delta_1 \times \tilde{\tau}_1 \\ \quad + \delta_2 \times \tilde{\tau}_2 \times f_{cf}, & FiTN_a^{e,\xi+1} < FiTN_a^{e,\xi} \\ \ddot{R}_a^{e,\xi}, & Otherwise \end{cases} \quad (22)$$

$$\begin{aligned} \tilde{\tau}_1 &= \ddot{\alpha}^{e,\xi} - \ddot{R}_{\delta_1}^{e,\xi} \\ \tilde{\tau}_2 &= \overline{\overline{CT}}^{e,\xi} - \ddot{R}_{\delta_2}^{e,\xi} \end{aligned} \quad (23)$$

Where, $\tilde{\tau}_1$ and $\tilde{\tau}_2$ refers to the alpha and pack influence, respectively, $FiTN_a^{e,\xi}$ represents the fitness of the coyote. Also, in equation (22), the term f_{cf} refers to the restriction factor that avoids the premature convergence problem and increases the global search ability of the algorithm and it is expressed as

$$f_{cf} = \frac{2.43 + \cos(2\pi/T_{IR} \times (\xi - T_{IR}/2))}{4} \quad (24)$$

Where, T_{IR} indicates the total number of iterations. Finally, the best coyote is selected based on its social condition obtained at the end of the iteration.

Step 2: Once optimal hyperparameters are selected, the following mathematical computations are performed in LSTM to get the network's output.

$$\ddot{G}\ddot{F}_\kappa = \psi^*(\omega^*_{\ddot{G}\ddot{F}} \cdot [\ddot{H}_{\kappa-1}, \overline{ISQ}_\kappa] + B^*_{\ddot{G}\ddot{F}}) \quad (25)$$

$$\ddot{G}\ddot{I}_\kappa = \psi^*(\omega^*_{\ddot{G}\ddot{I}} \cdot [\ddot{H}_{\kappa-1}, \overline{ISQ}_\kappa] + B^*_{\ddot{G}\ddot{I}}) \quad (26)$$

$$\ddot{G}\ddot{O}_\kappa = \psi^*(\omega^*_{\ddot{G}\ddot{O}} \cdot [\ddot{H}_{\kappa-1}, \overline{ISQ}_\kappa] + B^*_{\ddot{G}\ddot{O}}) \quad (27)$$

$$\begin{aligned} \ddot{G}\ddot{C}_\kappa &= \ddot{G}\ddot{F}_\kappa \ddot{G}\ddot{C}_{\kappa-1} + \ddot{G}\ddot{I}_\kappa \\ &\times \eta^*(\omega^*_{\ddot{G}\ddot{C}_\kappa} \cdot [\ddot{H}_{\kappa-1}, \overline{ISQ}_\kappa] + B^*_{\ddot{G}\ddot{C}_\kappa}) \end{aligned} \quad (28)$$

$$\ddot{H}_\kappa = \ddot{G}\ddot{O}_\kappa \times \eta^*(\ddot{G}\ddot{C}_\kappa) \quad (29)$$

Where, $\omega^*_{\ddot{G}\ddot{F}}$, $\omega^*_{\ddot{G}\ddot{I}}$, $\omega^*_{\ddot{G}\ddot{O}}$, and $\omega^*_{\ddot{G}\ddot{C}_\kappa}$ and $B^*_{\ddot{G}\ddot{F}}$, $B^*_{\ddot{G}\ddot{I}}$, $B^*_{\ddot{G}\ddot{O}}$, and $B^*_{\ddot{G}\ddot{C}_\kappa}$ refers to the optimal weight and biases of input ($\ddot{G}\ddot{I}$), output ($\ddot{G}\ddot{O}$), cell state ($\ddot{G}\ddot{C}$) and forget gates ($\ddot{G}\ddot{F}$), respectively, which are optimally selected using SCCOA. The term \overline{ISQ} denotes the input sequence at the time step κ , ψ^* and

η^* represents tangent and sigmoid activation function, respectively and \ddot{H}_κ indicates the output of the LSTM cell.

(a) SoftMax Layer

Finally, the SoftMax layer receives the output from the LSTM layer, and this layer classifies the input data into the depressed or non-depressed people.

4. Results and discussion

Here, the performance of the proposed DD of SM big data using an RNT-OLSTM is analyzed by comparing their results with the conventional schemes. First, the datasets used to analyze the proposed work's efficiency are described, and then the outcomes of the proposed and existing schemes are compared and discussed regarding some performance metrics. This system was implemented in the working platform of Python 3.7 with a machine configuration of Intel(R) Core-i7 CPU with a speed of 3.0 GHz, memory of 64 GB, and Window 10 OS.

4.1. Dataset description

Twitter is a well-known social network that offers open and straightforward data access. The experimentation is done by using two datasets such as clean_d_tweets.csv, dataset 1 (DS1), and CLPsych2015, dataset 2 (DS2). The DS 1 contains the depressed and non-depressed class labels of Twitter data from the users, which can be obtained via <https://www.kaggle.com/datasets/hyunkic/twitter-depression-dataset>, and it contains more than 4000 tweets. In this dataset, d_tweets represent depressed tweets, and non_d_tweets represent non-depressed tweets. Next, the DS2 is obtained via (https://github.com/clpsych/shared_task) which contains 2000 tweets of users to detect their depression level.

4.2. Performance evaluation of the techniques for clustering

In this section, the proposed SDFCM is compared against the existing FCM, Mean Shift (MS) clustering, Low-energy adaptive clustering hierarchy (LEACH), and Density Based Clustering (DBC) with respect to processing time metric, which is shown in Table 1.

Table 1. Clustering results analysis.

Methods	Processing time (min)
Proposed	0.82
FCM	1.05
MS	1.91
LEACH	2.38
DBC	3.14

The above table demonstrates the outstanding efficiency of the proposed work in terms of processing time. Processing times are important because they can help to estimate how long it will take to complete a clustering process. If the system takes minimal time to cluster the data, then it regarded as a good system. Herein also, the existing FCM, MS, LEACH, and DBC takes processing time of 1.05, 1.91, 2.38, and 3.14 min, which is higher when compared to the proposed one, because, the proposed system consumes very less time for clustering, i.e. it takes just 0.82 min. As in basic FCM algorithm, initial centroids are selected randomly from the input data, so clusters vary from one another, because of which the number of iterations and total processing time also increases in each run of the same data. In proposed SDFCM algorithm initial centroids are calculated by SD, so the number of iterations remains constant and

processing time is also decreased. This is the reason that proposed SDFCM clustering algorithm is efficient from basic FCM and MS, LEACH, and DBC algorithms.

4.3. Performance evaluation of the techniques for word embedding

This section compares the results of the proposed hybrid TF-IDF and Word2vec with the different existing word-embedding schemes such as TF-IDF, Word2vec, Glove, and Fast text regarding accuracy. Figure 3(a) shows the model’s results achieved for DS1 and DS2. We can easily observe that the proposed word embedding approach attains a better accuracy rate than the existing methods. For example, the existing Fast text has an accuracy of 88.32% and 88.04% for DS1 and DS2, and the Glove embedding approach has an accuracy

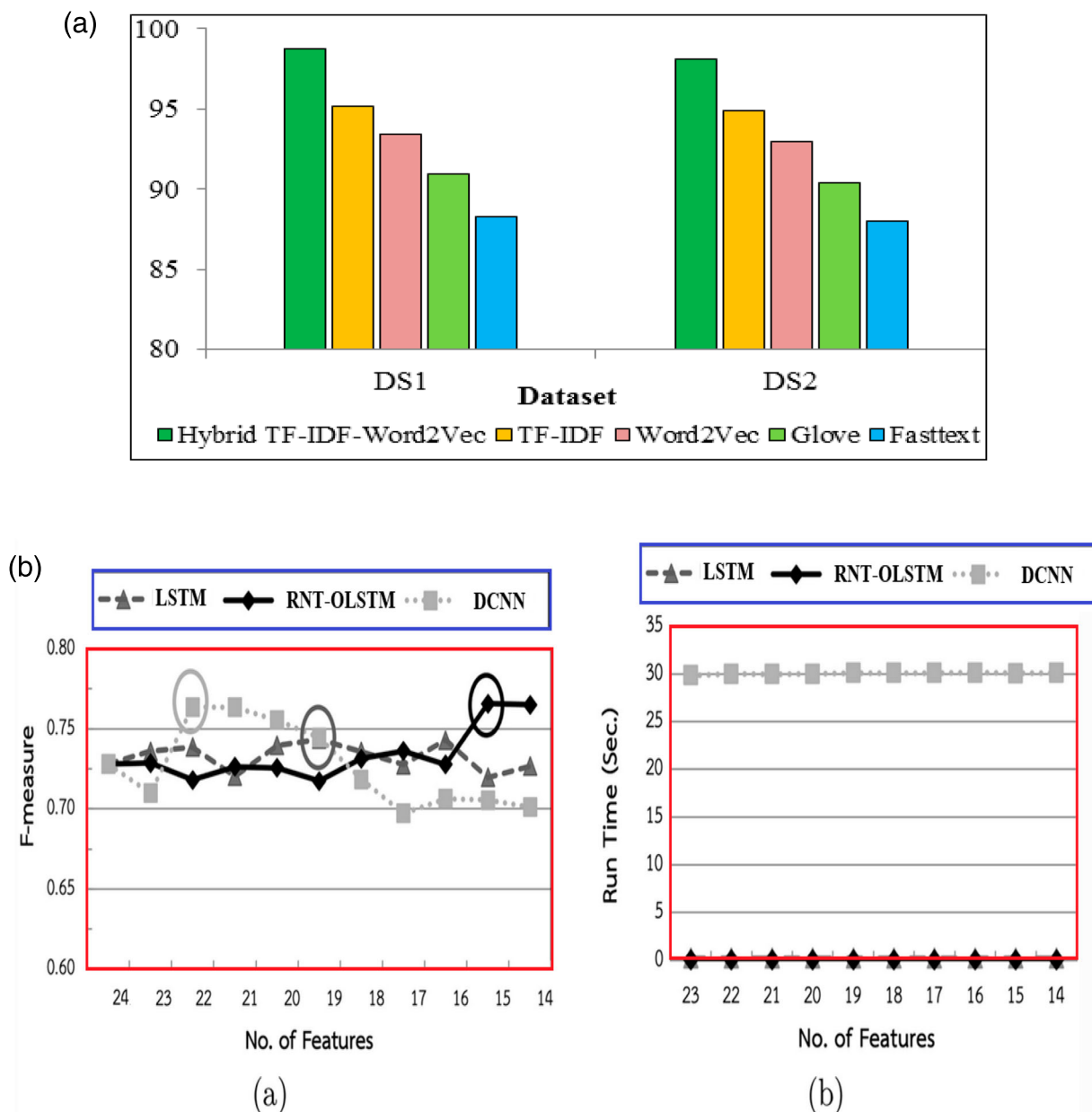


Figure 3. (a): Accuracy analysis of the proposed system. (b): F-measure and Run time comparison with proposed work.

of 90.96% and 90.45%. Also, the existing TF-IDF and Word2Vec approach used to embed the word alone has obtained the lowest accuracy of 95.14% and 93.47% for DS1 and 94.86% and 93.02% for DS2, but the proposed hybrid approach using TF-IDF and Word2Vec yields highest accuracy of 98.78% and 98.12% for DS1 and DS2. TF-IDF has a good representation of words that determine words' weight, but doesn't represent well the semantics while Word2Vec has opposing characteristics that is it can handle with problems related with semantics. By observing the advantages and disadvantages of their models, it is possible to presume that they complement each other. Combining these two models automatically increase the classification accuracy. This is the most important thing to achieve better outcomes for the proposed system. So, it is stated that using these hybrid embedding schemes represents the terms' vector representation better than the existing schemes and leads to higher accuracy.

4.4. Performance evaluation of the techniques for depression detection

This section investigates the outcomes of the proposed RNT-OLSTM with conventional LSTM, RF, Deep CNN (DCNN), and SVM for two datasets, namely DS1 and DS2. The evaluation includes metrics such as accuracy, f-measure, recall, precision, specificity, the area under the curve (AUC), and error rate. Table 1 demonstrates the efficiency of the proposed and existing methods for DS1. Figure 3(b) shows that proposed RNT-OLSTM comparison to other methods.

The existing SVM attains 91.48% accuracy, 90.81% precision, 91.59% recall, 91.18% precision, 85.57% specificity, 85.42% AUC score, and a 21.98 error rate. The existing RF has 93.58% accuracy, 92.91% precision, 93.69% recall, 93.28% f-measure, 90.47% specificity, 89.14% AUC, and 16.27 error rate, the existing DCNN has 95.26% accuracy, 94.76% precision, 95.38% recall, 95.01% f-measure, 93.41% specificity, 91.48% AUC, and 14.69 error rate, and the existing LSTM has 97.57% accuracy, 96.86% precision, 97.83% recall, 97.48% f-measure, 96.45% specificity, 93.21% AUC, and 7.26 error rate. The proposed RNT-OLSTM achieves 99.55% accuracy, 99.66% precision, 99.89% recall, 99.68% f-measure, 98.24% specificity, 96.25% AUC, and a 1.22 error rate.

In this table, the existing SVM has low performance than all the methods. For example, the existing SVM attains 91.48% accuracy, 90.81% precision, 91.59% recall, 91.18% precision, 85.57% specificity, 85.42% AUC score, and a 21.98 error rate, which are very low outcomes when compared to all the methods. But the proposed one achieves superior outcomes than all the existing methods. The reason is that, the existing LSTM, DCNN, RF, and SVM are the well-known algorithm and produce the satisfactory classification

Table 2. Performance evaluation of the models for DS1.

Metrics(%) / Techniques	Proposed				
	RNT-OLSTM	LSTM	DCNN	RF	SVM
Accuracy	99.55	97.57	95.26	93.58	91.48
Precision	99.66	96.86	94.76	92.91	90.81
Recall	99.89	97.83	95.38	93.69	91.59
F-measure	99.68	97.48	95.01	93.28	91.18
Specificity	98.24	96.45	93.41	90.47	85.57
AUC score	96.25	93.21	91.48	89.14	85.42
Error rate	1.22	7.26	14.69	16.27	21.98

Table 3. Performance evaluation of the models for DS2.

Metrics (%) / Techniques	Proposed				
	RNT-OLSTM	LSTM	DCNN	RF	SVM
Accuracy	99.23	97.32	95.06	93.29	91.18
Precision	99.38	97.45	95.17	93.42	91.32
Recall	99.58	97.61	95.32	93.64	91.52
F-measure	99.45	97.52	95.28	93.53	91.46
Specificity	97.87	95.85	92.74	89.83	84.18
AUC score	95.78	92.14	90.74	88.15	83.78
Error rate	1.68	9.08	13.14	14.25	18.41

results. However, these all the existing methods manually extracts feature from the embedded words for classification, but herein, the proposed system uses RNT method to efficiently extract the features and this efficient method leads to attain the higher performance than the existing methods. Also, the existing classification methods randomly choose the hyperparameter such as weight and bias for the corresponding inputs. This random selection of hyperparameter directly affects the classification performance and increases the error rate. But the proposed one addresses these issues and proposes a SCCOA approach to optimally select the hyperparameter. This optimal selection makes the proposed system faster and achieving high-level outcomes than the existing methods. For example, the proposed RNT-OLSTM's accuracy was improved by 1.98%, 4.29%, 5.97%, and 8.07% than the existing LSTM, DCNN, RF, and SVM methods. Also, the proposed one attains 99.66% precision, 99.89% recall, 99.68% f-measure, 98.24% specificity, 96.25% AUC, and a 1.22 error rate, which is also better than the existing methods. From Table 2, it is concluded that the proposed one achieves better outcomes than the conventional methods. Next, the outcomes of the detection models for DS2 are discussed, and their numerical outcomes are tabulated in Table 3.

Table 3 illustrates the efficiency of the proposed approach with existing methods using DS2. The proposed RNT-OLSTM attains an accuracy of 99.23%, higher than the existing methods. The existing SVM attains the lowest results for all metrics compared to other methods. It attains 91.32% precision, 91.52% recall, 91.46% f-measure, 84.18% specificity, 83.78% AUC, and 18.41 error rate, respectively. However, the proposed method achieves 99.38% precision, 99.58% recall, 99.45% f-measure, 97.87% specificity, 95.78% AUC, and a 1.68 error rate, which is better than LSTM,

DCNN, RF, and SVM. The results prove the effectiveness of the proposed model over existing methodologies. The reason is that the proposed one uses the SDFCM-SMOTE approach to address the data imbalance problem, which leads to higher predictive accuracy for DD. When the data is not balanced with both majority and minority classes, the classifiers use the biased learning model that could lead to poor predictive accuracy for the minority classes contrasted to the majority classes. However, the existing methods are focused on something other than the data imbalance problem, so they attain minimal outcomes than the proposed one. In addition, tuning the hyperparameters in the DL model is essential because the random selection increases the classifier's training time and error rate. So, the proposed work uses SCCOA for the optimal selection of hyperparameters in the LSTM model, which makes the proposed one achieve better detection results with a minimal error rate. The entire experimental findings, therefore, demonstrated that the proposed strategy works better than current approaches.

5. Conclusion

In this paper, the proposed system uses a novel DL framework for the DD of prominent SM data. The proposed work contains “4” phases, namely, data balancing, pre-processing, word embedding, and DD. The experimentation is done on two openly available datasets, and the outcomes of the proposed work are investigated in two ways. First, the outcome of the proposed word embedding technique, namely, hybrid TF-IDF-Word2Vec, is analyzed with the conventional TF-IDF, Word2Vec, Glove, and Fast text. Here, the proposed system achieves 98.78% and 98.12% accuracy for DS1 and DS2. Next, the proposed RNT-OLSTM is weighted against the conventional LSTM, DCNN, RF, and SVM approaches regarding the accuracy, f-measure, recall, precision, specificity, AUC, and error rate. Herein, the proposed one achieves the highest accuracy of 99.55% and 99.23% with minimal error rates of 1.22 and 1.68 for both datasets. Thus, the overall outcomes showed that the proposed one is superior to the existing state-of-the-art techniques. In future, this work will be prolonged using an advanced DL approach to send the degree of depression to the medical team or relatives of the patient.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

- [1] Ghosh S, Anwar T. Depression intensity estimation via social media: a deep learning approach. *IEEE Trans Comput Social Syst.* 2021;8(6):1465–1474. doi:10.1109/TCSS.2021.3084154
- [2] Aragon ME, Lopez-Monroy AP, Gonzalez-Gurrola LCG, et al. Detecting mental disorders in social media through emotional patterns—the case of anorexia and depression. *IEEE Trans Affect Comput.* 2021.
- [3] Lara JS, Aragón ME, González FA, et al. Deep bag-of-sub-emotions for depression detection in social media. In: *Text, Speech, and Dialogue: 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6–9, 2021, Proceedings 24.* Springer International Publishing; 2021. p. 60–72.
- [4] Salas-Zárate R, Alor-Hernández G, Salas-Zárate MDP, et al. Detecting depression signs on social media: A systematic literature review. *Healthcare.* 2022;10(2):291, doi:10.3390/healthcare10020291
- [5] Rustagi A, Manchanda C, Sharma N, et al. Depression anatomy using combinational deep neural network. In: *International conference on innovative computing and communications: proceedings of ICICC 2020, Volume 2.* Springer Singapore; 2021. p. 19–33.
- [6] Ye J, Yu Y, Wang Q, et al. Multi-modal depression detection based on emotional audio and evaluation text. *J Affect Disord.* 2021;295:904–913. doi:10.1016/j.jad.2021.08.090
- [7] Angskun J, Tipprasert S, Angskun T. Big data analytics on social networks for real-time depression detection. *J Big Data.* 2022;9(1):69, doi:10.1186/s40537-022-00622-2
- [8] Govindasamy KA, Palanichamy N. Depression detection using machine learning techniques on twitter data. In: *2021 5th international conference on intelligent computing and control systems (ICICCS).* IEEE; 2021. p. 960–966.
- [9] William D, Suhartono D. Text-based depression detection on social media posts: A systematic literature review. *Procedia Comput Sci.* 2021;179:582–589. doi:10.1016/j.procs.2021.01.043
- [10] Smys S, Raj JS. Analysis of deep learning techniques for early detection of depression on social media network—a comparative study. *J Trends Comput Sci Smart Technol (TCSST).* 2021;3(01):24–39. doi:10.36548/jtcsst.2021.1.003
- [11] Haque A, Reddi V, Giallanza T. Deep learning for suicide and depression identification with unsupervised label correction. In: *Artificial Neural Networks and Machine Learning—ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part V 30.* Springer International Publishing; 2021. p. 436–447.
- [12] Jain V, Kumar V, Pal V, et al. Detection of cyberbullying on social media using machine learning. In: *2021 5th international conference on computing methodologies and communication (ICCMC).* IEEE; 2021. p. 1091–1096.
- [13] Kumar P, Chauhan R, Stephan T, et al. A machine learning implementation for mental health care. Application: smart watch for depression detection. In: *2021 11th international conference on cloud computing, data science & engineering (confluence).* IEEE; 2021. p. 568–574.
- [14] Bathina KC, Ten Thij M, Lorenzo-Luaces L, et al. Individuals with depression express more distorted thinking on social media. *Nature Human Behaviour.* 2021;5(4):458–466. doi:10.1038/s41562-021-01050-7
- [15] Malviya K, Roy B, Saritha SK. A transformers approach to detect depression in social media. In: *2021 international conference on artificial intelligence and smart systems (ICAIS).* IEEE; 2021. p. 718–723.

- [16] Joshi ML, Kanoongo N. Depression detection using emotional artificial intelligence and machine learning: A closer review. *Mater Today Proc.* 2022;58:217–226. doi:10.1016/j.matpr.2022.01.467
- [17] Ortega-Mendoza RM, Hernández-Farías DI, Montes-y-Gómez M, et al. Revealing traces of depression through personal statements analysis in social media. *Artif Intell Med.* 2022;123:102202, doi:10.1016/j.artmed.2021.102202
- [18] Kaur H, Ahsaan SU, Alankar B, et al. A proposed sentiment analysis deep learning algorithm for analyzing COVID-19 tweets. *Inform Syst Front.* 2021: 1–13.
- [19] Uddin MZ, Dysthe KK, Følstad A, et al. Deep learning for prediction of depressive symptoms in a large textual dataset. *Neural Comput Applic.* 2022;34(1):721–744. doi:10.1007/s00521-021-06426-4
- [20] Loh HW, Ooi CP, Aydemir E, et al. Decision support system for major depression detection using spectrogram and convolution neural network with EEG signals. *Expert Syst.* 2022;39(3):e12773, doi:10.1111/exsy.12773
- [21] Tong L, Liu Z, Jiang Z. Cost-sensitive boosting pruning trees for depression detection on twitter. *IEEE Trans Affect Comput.* 2022.
- [22] Amanat A, Rizwan M, Javed AR, et al. Deep learning for depression detection from textual data. *Electronics (Basel).* 2022;11(5):676, doi:10.3390/electronics11050676
- [23] Safa R, Bayat P, Moghtader L. Automatic detection of depression symptoms in twitter using multimodal analysis. *J Supercomput.* 2022;78(4):4709–4744. doi:10.1007/s11227-021-04040-8
- [24] Chiong R, Budhi GS, Dhakal S. Combining sentiment lexicons and content-based features for depression detection. *IEEE Intell Syst.* 2021;36(6):99–105. doi:10.1109/MIS.2021.3093660
- [25] Zogan H, Razzak I, Wang X, et al. Explainable depression detection with multi-aspect features using a hybrid deep learning model on social media. *World Wide Web.* 2022;25(1):281–304. doi:10.1007/s11280-021-00992-2
- [26] Chiu CY, Lane HY, Koh JL, et al. Multimodal depression detection on Instagram considering time interval of posts. *J Intell Inf Syst.* 2021;56:25–47. doi:10.1007/s10844-020-00599-5
- [27] Chiong R, Budhi GS, Dhakal S, et al. A textual-based featuring approach for depression detection using machine learning classifiers and social media texts. *Comput Biol Med* 2021;135:104499, doi:10.1016/j.combiomed.2021.104499
- [28] Zogan H, Razzak I, Jameel S, et al. (2021). Depression-Net: A novel summarization boosted deep framework for depression detection on social media. *arXiv preprint arXiv:2105.10878.*
- [29] Nijhawan T, Attigeri G, Ananthakrishna T. Stress detection using natural language processing and machine learning over social interactions. *J Big Data.* 2022;9(1): 1–24. doi:10.1186/s40537-022-00575-6
- [30] Chatterjee R, Gupta RK, Gupta B. Depression detection from social media posts using multinomial naive theorem. In: *IOP conference series: materials science and engineering (Vol. 1022, No. 1, p. 012095).* IOP Publishing; 2021.
- [31] Sundaram V, Ahmed S, Muqtadeer SA, et al. Emotion analysis in text using TF-IDF. In: *2021 11th international conference on cloud computing, data science & engineering (confluence).* IEEE; 2021. p. 292–297.
- [32] Wani MA, ELAffendi MA, Shakil KA, et al. Depression screening in humans with AI and deep learning techniques. *IEEE Trans Comput Soc Syst.* 2022.
- [33] Lin B-S, Lee I-J, Wang S-P, et al. Residual neural network and long short-term memory-based algorithm for estimating the motion trajectory of inertial measurement units. *IEEE Sensors J.* 2022;22(7):6910–6919. doi:10.1109/JSEN.2022.3153398