Design And Analyze the Framework for Preventing Cyberbullying in Social Networking Sites Using a Deep Learning Model

Balaji Nallapati [1], Pawan Kumar [1], M. Latha [2]

[1]*Student, Department of Computer science and Engineering, Adithya institute of technology, Coimbatore, Anna University, Tamil Nadu, India.*

[2]*Supervisor, Department of Computer science and Engineering, Adithya institute of technology, Coimbatore, Anna University, Tamil Nadu, India.*

Corresponding Author: balajibalu5210@gmail.com

Abstract: - Cyberbullying is bullying that takes place over digital devices like cell phones, computers, and tablets. Cyberbullying can occur through SMS, Text, and apps, or online in social media, forums, or gaming where people can view, participate in, or share content. Cyberbullying includes sending, posting, or sharing negative, harmful, false, or mean content about someone else. It can include sharing personal or private information about someone else causing embarrassment or humiliation. The content an individual share online – both their personal content as well as any negative, mean, or hurtful content – creates a kind of permanent public record of their views, activities, and behavior. To avoid or detecting cyberbullying attacks, many existing approaches in the literature incorporate Machine Learning and Natural Language Processing text classification models without considering the sentence semantics. The main goal of this project is to overcome that issue. This project proposed a model LSTM - CNN architecture for detecting cyberbullying attacks and it used word2vec to train the custom of word embeddings. This model is used to classify tweets or comments as bullying or non-bullying based on the toxicity score. LSTM networks are well-suited to classifying, processing, and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. A convolutional neural network (CNN) is a type of artificial neural network and it has a convolutional layer to extract information by a larger piece of text and by using this model LSTM- CNN achieve a higher accuracy in analysis, classification and detecting the cyberbullying attacks on posts and comments.

Key Words— ***Cyber-bullying, tweet classification, Dolphin Echolocation algorithm, Elman recurrentneural networks, short text topic modeling, cyberbullying detection, social media.***

I. INTRODUCTION

Social media networks such as Facebook, Twitter, Flickr, and Instagram have become the preferred online platforms for interaction and socialization among people of all ages.

Manuscript revised May 09, 2023; accepted May 13, 2023. Date of publication May 10, 2023.

This paper available online at www.ijprse.com

ISSN (Online): 2582-7898; SJIF: 5.59

While these platforms enable people to communicate and interact inpreviously unthinkable ways, they have also led to malevolent activities such as cyber-bullying. Cyberbullying is a type of psychological abuse with a significant impact on society. Cyber-bullying events have been increasing mostly among young people spending most of their time navigating between different social media platforms. Particularly, social media networks such as Twitter and Facebook are prone to CB because of their popularity and the anonymity that the Internet provides to abusers. In India, for example, 14 percent of all provides to abusers. In India, for example, 14 percent of all harassment occurs on Facebook and Twitter, with 37 per-cent of these incidents involving youngsters. Moreover, cyberbullying might lead to serious mental issues and adverse mental health effects. Most suicides are due to the anxiety, depression, stress, and social and emotional difficulties from

cyber-bullying events. This motivates the need for anapproach to identify cyberbullying in social media messages(e.g., posts, tweets, and comments).

In this article, we mainly focus on the problem of cyber-bullying detection on the Twitter platform. As cyberbullying is becoming a prevalent problem in Twitter, the detection of cyberbullying events from tweets and provisioning preventive measures are the primary tasks in battling cyberbullying threats. Therefore, there is a greater need to increase the research on social networks-based CB in order to get greater insights and aid in the development of effective tools and approaches to effectively combat cyberbullying problem. Manually monitoring and controlling cyberbullying on Twitter platform is virtually impossible. Furthermore, mining social media messages for cyberbullying detection is quite difficult. For example, Twitter messages are often brief, full of slang, and may include emojis, and gifs, which makes it impossible to deduce individuals' intentions and meanings purely from social media messages. Moreover, bullying can be difficult to detect if the bully uses strategies like sarcasm or passive-aggressiveness to conceal it.

Despite the challenges that social media messages bring, cyberbullying detection on social media is an open and active research topic. Cyberbullying detection within the Twitter platform has largely been pursued through tweet classification and to a certain extent with topic modeling approaches.Text classification based on supervised machine learning (ML) models are commonly used for classifying tweets intobullying and non-bullying tweets. Deep learning (DL) based classifiers have also been used for classifying tweets into bullying and non-bullying tweets. Supervised classifiers have low performance in case the classlabels are unchangeable and are not relevant to the new events. Also, it may be suitable only for a pre-determined collection of events, but it cannot successfully handle tweetsthat change on the fly. Topic modeling approaches have longbeen utilized as the medium to extract the vital topics froma set of data to form the patterns or classes in the complete dataset. Although the concept is similar, the general unsupervised topic models cannot be efficient for short texts, and hence specialized unsupervised short text topic models wereemployed. These models effectively identify the trending topics from tweets and extract them for further processing. These models help in leveraging the bidirectional processingto extract meaningful topics. However, these unsupervised models require extensive training to obtain sufficient prior knowledge, which is not adequate in all cases. Considering these limitations, an efficient tweet classification approach must be developed to bridge the gap between the classifier and the topic model so that the adaptability is significantly proficient.

In this article, we propose a hybrid deep learning-based approach, called DEA-RNN, which automatically detects bullying from tweets. The DEA-RNN approach combines Elman type Recurrent Neural Networks (RNN) with an improved Dolphin Echolocation Algorithm (DEA) for fine-tuning the Elman RNN's parameters. DEA-RNN can handle the dynamic nature of short texts and can cope with the topic models for the effective extraction of trending topics. DEA-RNN outperformed the considered existing approaches in detecting cyberbullying on the Twitter platform in all scenarios and with various evaluation metrics. The contributions of this article can be summarized as the following:

- Develop an improved optimization model of DEA for use to automatically tune the RNN parameters to enhance the performance;
- Propose DEA-RNN by combining the Elman type RNN and the improved DEA for optimal classification of tweets;
- A new Twitter dataset is collected based on cyber-bullying keywords for evaluating the performance of DEA-RNN and the existing methods;
- The efficiency of DEA-RNN in recognizing and classifying cyberbullying tweets is assessed using Twitter datasets. The thorough experimental results reveal that DEA-RNN outperforms other competing modelsin terms of recall, precision, accuracy, F1 score, and specificity.

The rest of this article is structured as the following: Recent related works are reviewed and analyzed in Section II. The proposed DEA-RNN model is described in Section III. Section IV discusses the experimental analysis, performance metrics, and results analysis. The discussion is introduced in Section V. Finally, Section VI offers the conclusion and possible future directions

II. Related Works

This section is mainly focused on reviewing state-of-the-art of CB detection and classification on Twitter datasets. Machine learning (ML) based approaches with different feature selection methods are widely used in cyberbullying

tweet classification. Purnamasari *et al.* [26] utilized the SVM and Information Gain (IG) based feature selection method for detecting cyberbullying events in tweets. Muneer and Fati [11] used various classifiers, namely AdaBoost (ADB), Light Gradient Boosting Machine (LGBM), SVM, RF, Stochastic Gradient Descent (SGD), Logistic Regression (LR), and MNB, and for cyberbullying events identificationin tweets. This study extracted features using Word2Vec and TF-IDF methods. Dalvi *et al.* [12] [27] used SVM and Ran dom Forests (RF) models with TF-IDF for feature extraction for detecting cyberbullying in tweets. Although SVMin these models achieved high performance, the model complexity increases when the class labels are increased. Algaradi *et al.* [28] investigated cyberbullying identificationusing different ML classifiers such as RF, Naïve Bayes(NB), and SVM based on various extracted features from Twitter such as (tweet content, activity, network, and user). Huang *et al.* [29] suggested an approach for identifying CB from social media, which integrated the social media features and textual content features. The features are ranked using IG method. Well-known classifies such as NB, J48, and Bagging and Dagging are utilized. The findings impliedthat social characteristics could aid in increasing the accu- racy of cyberbullying detection. Squicciarini *et al.* [30] uti- lized a decision tree (C4.5) classifier with a social network, personal and textual features to identify Cyberbullying and cyberbullying prediction on social networks like spring.me, and MySpace. Balakrishnan *et al.* [31] utilized different MLalgorithms such as RF, NB, and J48 to detect cyberbullying events from tweets and classify tweets to different cyber- bullying classes such as aggressors, spammer, bully, and normal. The study concluded that the emotional feature doesnot impact the detection rate. Despite its efficiency, thismodel is limited to a small dataset with fewer class labels.Alam *et al.* [32] proposed an ensemble-based classificationapproach using the single and double ensemble-based voting model. These ensemble-based voting models utilized decision tree, LR, and Bagging ensemble model classifiers forthe classification while utilizing mutual information bigramsand unigram TF-IDF as feature extraction models. On anal ysis over the Twitter dataset, the Bagging ensemble modelprovided the best precision but considered other parameters. Although, these ensemble models reduced the training andexecution time for classification, the major limitation comes when utilized sarcasm tweets and multiple-meaning acronymterms. Chia *et al.* [8] also utilized different ML and feature engineering-based approaches to classify irony and sarcasm from cyber-bullying tweets. In this approach, many classifiers and feature selection methods were tested; while this approach greatly detects the sarcasm and irony terms amongcyber-bullying tweets, the detection rate is still very low [33].Similarly, Rafiq *et al.* [17] utilized decision tree, AdaBoost, NB, and Random Forest classifier to identify the instances of cyberbullying in a Vine dataset. Authors collected the Vine media dataset and labeled it using Crowd-Sourcedand CrowdFlower websites. They utilized the comments, unigrams, media information, and profile as the features. Nahar *et al.* [34] suggested a semi-supervised learningmethod for detecting CB in social media in which trainingdata samples are augmented, and a fuzzy SVM methodis applied. The augmented training approach expands and extracts the training set from the unclassified streamingtext automatically. The learning is performed using a small limited training set given as an initial input. The suggested method overcomes the dynamic and complex character of streaming data. Xu *et al*. [35] provided many off-the-shelf methods, including LDA and LSA-based modeling and Bag-of-Words models for predicting bullying traces on Twitter. A personalized cyberbullying detection framework, namely PI-Bully, was introduced by Cheng *et al.* [36] to detect cyber-bullying from the Twitter dataset. PI-Bully composes three elements: a global element that determines the characteristics that all users have in common, a personalized element that captures the distinctive features of each user, and a peer influence element capable of quantifying the various influences of other users people. Deep learning (DL) based approaches for cyberbullying detection in tweets have also been proposed in the literature. N. Yuvaraj *et al.* [9] used Artificial Neural Net-work (ANN) and Deep Reinforcement Learning (DRL) to classify cyberbullying tweets. However, this approach has higher computational complexity. Chen *et al.* [37] used a text classification model based on CNN and 2-D TF-IDF features to enhance the sentiment analysis task performance. The experimental results showed that the CNN model obtained optimal results compared to the baselines LR and SVM models. Agrawal [16] utilized LSTM with Transfer Learning for cyberbullying detection on several social media networks. A new representation learning approach named smSDA (Semantic-Enhanced Marginalized Denoising Autoencoder)was suggested by Zhao *et al.* [38] for detecting cyberbullying. smSDA produced discriminative and robust representations. Following that, the numerical representations that have been learned can be input into SVM. Zhang [39] Suggested a new model which integrates the Gated Recur-

rent unit Network GRU layers and CNN layers to detect hate speech. Al-Hassan and Al-Dossari [19] utilized SVM as the baseline classifier and compared it against four DL models, namely CNN LTSM, LTSM, CNN GRU, and GRU to detect cyberbullying hate speech in Arabic tweets. However, the CNN LSTM and CNN GRU complexity is higher and might not be effective in handling larger datasets. Natarajan Yuvaraj *et al.* [18] proposed a new classification model for CB detection from Twitter data. It used deep decision-tree classification with multifeature based AI for tweet classification. The deep decision tree classifier has been designed by integrating the hidden layers of deep neural networks with the decision tree classifier. This approach also utilized three feature selection approaches: Chi-Square, Pearson Correlation, and IG. However, it cannot handlehigh-dimensional data with such accuracy. Fang *et al.* [20] designed a classification model that combines a self-Attention mechanism and bi-directional Gated RecurrentUnit (Bi-GRU) to detect cyberbullying in tweets. This model employed merit for learning the underlying relationships between words using BI-GRU and used it together with a self-attention mechanism to improve the cyberbullying tweets classification process. However, the context-independent behavior of the attention network creates limitations in learning all relationships between the tweets. Pericherla and Ilavarasan [33] suggested a transformer network-based word embedding model to classify CB tweets.This model utilizes Light Gradient Boosting Machine to classify the tweets and Robert a to create word embed- ding. This approach overcomes the context-independent limitations of traditional word embedding methods. Yet, this model has a higher training time compared to the CNN models. Paul and Saha [40] proposed a model for identifying cyberbullying, namely CyberBERT, based on the BERT. Iwendi *et al.* [21] introduced a model to detect cyber- bullying based on Bi-LSTM and RNN. This model showed that the RNN could achieve high performance, but still, the Bi-LSTM has significantly high efficiency. In some cases, CNN also performs better. Akhter *et al.* [41] performed many DL models such as LSTM, CLSTM, CNN, and BLSTM, and other ML models to discover an abusive language from Urdu social media text. Some other studies utilized CNN's to enhance the cyberbullying detection [42]–[46]. Tripathy *et al.* [47] proposed a fine-tuning approach for detecting CB based on ALBER. Agarwal *et al.* [7] utilized RNN based on Under-Sampling and Class Weighting. Thesemodifications helped the RNN model to perform better thanthe LSTM model. This indicates that tuning the parameters can enhance the RNN performance. Pitsilis *et al.* [48] proposed hate-speech detection utilizing RNN and the word frequency vectors. Edo-Osagie *et al.* [49] developed Attention-based RNN for short text classification and achieved high accuracy. However, the location filtering in this method is limited. Khodabakhsh *et al.* [50] presented future personal life events predictions from tweets using theRNN model. However, this model does not classify the highly class-imbalanced data effectively. Kumar and Sachdeva [51] proposed a hybrid approach to detect CB in social media. This approach integrates the capsule network (CapsNet) and Bi-GRU encoder, namely (Bi-GAC). Cheng *et al.* [52] sug- gested an approach, namely HANCD (Hierarchical Atten- tion Network for Cyberbullying Detection). The proposed approach utilized the context to detect the relative signifi- cance of the specific comments and words by applying the levels of attention techniques. Besides, it forecasts the time interval that elapses between two neighboring comments. Eronen *et al.* [53] suggested an approach for detecting cyber-bullying based on the linguistically backed pre-processing and Feature Density (FD) approach. The authors investigatedthe effectiveness of FD utilizing linguistically-backed pre- processing such as stop words filtering, Parts of Speech (POS), Named Entity Recognition (NER), etc., approaches for assessing classification performance and the complexity of the dataset. On the other side, some recent studies pre- sented multi-models to detect CB in 3 various modalities of social data networking, namely visual and info-graphic and textual such as [51], [54], [55]. Kumari *et al.* [56] presentedDL based model to classify various levels of cyber aggression over networking social media comments in a bilingual. From the above-detailed review of the literature related toCB detection and classification, some important issues havebeen observed. Firstly, the deep learning classifiers have bet-ter classification efficiency than the machine learning modelsbecause of their superiority in terms of accuracy when it getstrained with a large dataset. Secondly, RNN has better advan-tages of fast processing with the abstract feature learning process, thus making RNN as one of the most efficient classi- fication models. However, the limitations of the RNN modelare also highlighted, such as low accuracy due to pre-mature convergence, and limited tuning of RNN parameters have a significant impact on the overall classification performance. This indicates that tuning the parameters can enhance the RNN performance. Therefore, in this paper, the DEA-RNN model is presented to enhance the performance of RNN by

considering the aforesaid issues and limitations of existing ML and DL methods.

III. METHODOLOGY

The overall DEA-RNN model is shown in Fig. 1. The model includes the following phases: (i) data collection, (ii) data annotation, (iii) pre-processing and data cleansing, (iv) feature extraction and feature selection, and (v) classification. In the following subsection, each of these components are highlighted.

3.1 Data Collection

The input dataset is made up of tweets collected through Twit- ter API streaming with the help of around 32 cyber-bullying keywords. Idiot, ni**er, LGBTQ (le***an, g*y, bisexual, transgender, and queer), whore, pussy, faggot, shit, sucker, slut, donkey, live, afraid, moron, poser, rape, fuck, fucking, ugly, bitch, ass, whale, etc. are some of the keywords as recommended in psychology literature [30], [36], [57]. Whereasthe other keywords such as ban, kill, die, evil, hate, attack, terrorist, threat, racism, black, Muslim, Islam, and Islamic weresuggested in [39].

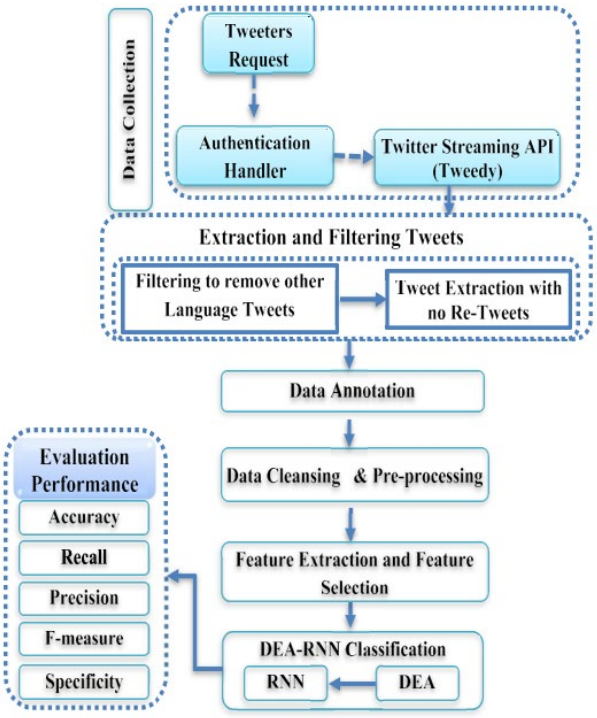

Fig.1. Methodology of the proposed model.

Table.1. The details of twitter dataset versions

Dataset	Total number of tweets	Number of Cyberbullying Tweets	# of Non-Cyberbullying Tweets
Original Twitter	10,000	3,492	6,508
Oversampled Twitter	13016	6,508	6,508

The initial dataset includes 435764 with racism, insult, swear, and sexism words-based keywords con-tributing about 130000 tweets. Tweets in this dataset includemany outliers. Only the English language tweets are needed, and hence the tweets containing other language terms are removed, and retweets are filtered, as shown in Fig.1. After removing these types of irrelevant tweets, about 10000 tweets are randomly selected from the remaining tweets to form thefinalized dataset. All these processes are done as a part of thepre-processing stage automatically. Then the other primary pre-processing operations are performed as in section III-C. after oversampling was 13,016 samples. Table 1 shows the original dataset and the dataset with oversampling.

3.2 Data Annotation

This section mainly concentrates on annotating and labeling the selected tweets from the original Twitter dataset. After selecting 10000 tweets randomly from the collecting tweets, the selected tweets were labeled manually into two labels, either ‘‘0’’ non-cyber bullying or ‘‘1’’ cyberbullying, by a set of three human annotators over a period ofone and half months. In the labeling procedure, the human annotators labeled the instances based on whether it was considered to involve cyberbullying and also the guidelines described in detail in [57]. The making decision of the cyberbullying instances depends on the following guidelines: character attacks, insults, competence attacks, malediction, verbal abuse, teasing, name-calling, mockery, threats, and physical appearance. Initially, each tweet was classified by two annotators, and the level of agreement rate between the two annotators was 91% approximately at this phase. Then, a third annotator was tasked with resolving the discrepancies discovered during the initial annotation process. Finally,we obtained the final dataset after resolving discrepancies and cleaned up the data, which contained 10000 labeled tweets, among which 6,508 (0.65%) are non-cyberbullying, and 3492 (0.35%) are cyberbullying tweets. By observing the number of cyberbullying and non-cyberbullying tweets, the labeled Twitter dataset is imbalanced. The number of tweets in

classes is greatly variable. As a result, balancing approaches such as oversampling or under-sampling is employed to resolve the issue. Here, Synthetic Minority Oversampling Technique (SMOTE) has been utilized to over-sample the minority class (cyberbullying Tweets) due to the class imbalance problem between cyberbullying and non-cyberbullying. The oversampling process is performed by replicating cyberbullying samples many times to balance the dataset as used in [15], [16]. Hence, the total number of tweets

3.3 Pre-Processing And Data Cleansing

The data cleansing and pre-processing phase contain three sub-phases [58]. This process is performed on the raw tweet dataset to form the finalized data as described inthe previous dataset. In the first sub-phase, noise removal such as URL removal, hashtag/mentions removal, punctuation/symbol removal, and emoticon transformation processesare performed. In the second sub-phase, Out of Vocabulary Cleansing such as spell checking, acronym expansion, slang modification, elongated (repeated Characters removal) are performed. In the final sub-phase, tweet transformations such as lower-case conversion, stemming, word segmentation (tok- enization), and stop word filtering are conducted. These sub- phases are performed to enhance the tweets and improve feature extraction and classification accuracy. Figure 2 shows the pre-processing and data cleansing steps.

3.4 Feature Extraction and Selection

The features from the Twitter dataset are extracted using NLP tools such as Word2Vec and TF-IDF, with the nouns, pronouns, and adjectives are considered as primary feature contents, whereas the adverbs and verbs provide additional information. Furthermore, the extraction of Part-of-Speech (POS) tags, function words, and content word features can improve the classification performance [59]. There are so many Feature selection methods as mentioned in [60]. For identifying the cyber-bullying events, prominent feature is selected utilizing the Information Gain (GI) method, then these features subsets are fed into DEA-RNN classifier.

3.5 DEA-RNN Classifier Model

1) IMPROVED DEA

DEA mimics the behaviors and the capability of dolphins to generate a kind of echo (click sounds) during the hunting process [61]. Initially, the dolphin's population is initial- ized, and the search space alternatives for each feature are ordered in ascending or descending order. For variable j, feature vectors A_j with the length LA_j is constructed, which includes all potential alternatives for the *jth* variable.

These vectors are then placed adjacent to each other and creating $alternatives_{NL\ \ NV}$ matrix as the columns in the alternative matrix based on this sorting process. Where the dimension of every location (Number of Variables) is denoted by NV,and the number of locations is denoted by NL. The dolphins'Number of Locations (NL locations) is then chosen at random in reasonable way, and the change in Convergence Factor (CF) is decided by the current loop's Predefined probability(PP) computation as expressed in Eq. (1).

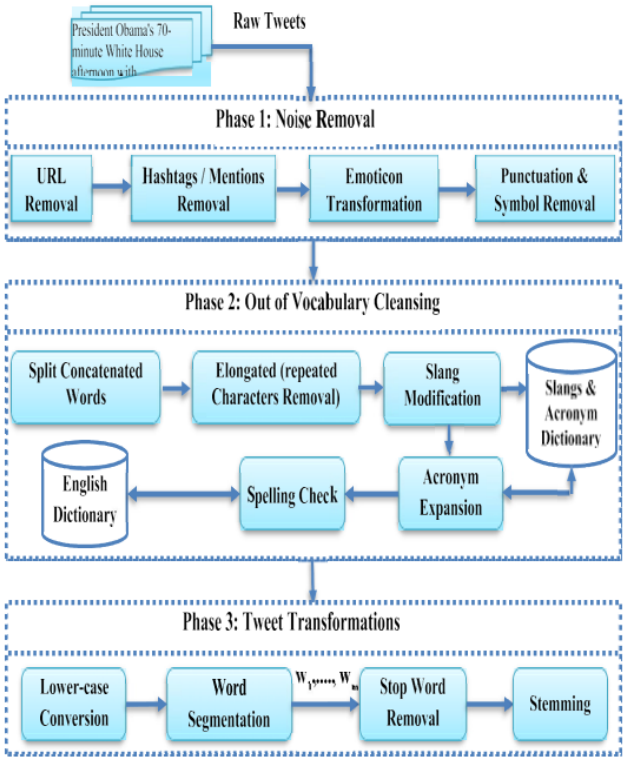

Fig.2. Data cleansing and pre-processing steps.

$$pp\ (Loop_i) = PP_1 + (1 - PP_1) \times A \qquad (1)$$

where

$$A = \frac{Loop_i^{power} - 1}{(Loops\ Number)^{power} - 1,}$$

The predefined probability is referred to PP, the CF of the 1st loop is denoted by PP_1, the current loop number is referredto $Loop_i$, $Loops\ number$ indicates the number of the loops thatthe algorithm considers for converging. The curve degree isdenoted by $Power$.

The fitness of each location is calculated using the errorrate equation with a threshold value of 0.57. The Accumulative

Fitness $AF_{(A+kj}$ is then calculated based on the rules of dolphin for jth variable, and ith location and $k = -R_e$ to R_e.

$$AF_{(A+kj} = Coeff\,(k) \times Fitness\,(i) + AF_{(A+kj}\,(2)\,|k|\cdot \quad (3)$$

R_e)

where, $_{AF(A+k)j}$ denotes to the

Accumulative Fitness of the $(A + k)$th alternative to be selected for the jth variable, the fitness in location i is denoted by $Fitness\,(i)$, R_e denotes the effective radius where its fitness affects the accumulative fitness of alternative A's neighbors and the radius should be no more than a quarter of the search space. Eq. (2) and (3) are modified to tweak the performance adaptability to the RNN. The $Coeff\,(k)$ is altered from a bi-linear coefficient function into a non-linear function as in Eq. (4), enabling the Dolphins to move in any direction within the search space of features. The non-linear nature coefficient function allows the matching of features with less iteration and also enhances the exploration process.

$$Coeff\,(k) = 1 - \frac{\sqrt{R_e^2\,(|k| - R_e)}^2}{R_e} \quad (4)$$

Using the modifying Coeff (k) as in Eq. (4), the AF(A+k) j Accumulative fitness as presented in Eq. (2) in DEA is altered and identified as in Eq. (5)

$$AF_{(A+k)j} = \left(1 - \frac{\sqrt{R_e^2\,(|k| - R_e)}^2}{R_e}\right) \times Fitness\,(i) + AF_{(A+k)j} \quad (5)$$

A small value of ε should be appended to the matrices in order to distribute the possibilities much fairly in the search space, as AF = AF + ε. This value has to be selected based on the way of defining the fitness function. Then, the optimal position of the current loop is detected and set AF = 0. For the variable j(j=1 to NV) , the probability (Pij) of the selecting alternative i(i=1 to ALj) is computed as shown in Eq. (6).

$$Coeff(k) = (1 -$$

$$\overline{P_{ij}} = \frac{AF_{ij}}{\sum_{i=1}^{Alj} AF_{ij}} \quad (6)$$

where Alj is the number of alternatives. Finally, the alternatives selected for all the variables with the best locations are specified with probability equal to PP as in the following formula: Pij = PP, whereas the remaining of probability is specified with other alternatives as given in Eq. (7).

$$P_{ij} = (1 - PP)\,\overline{P_{ij}} \quad (7)$$

This kind of probability can assist in identifying the following step locations, and lastly, the optimal global location is chosen. According to the algorithm's mapping, this position is the highest-rank configuration of RNN. By using the DEA, the training time of RNN can be reduced. As RNN is the widely utilized tool for classification, the slow speed of convergence limitation is primarily considered a problem that can be resolved using parameter optimization.

2) DEA-RNN WITH PARAMETER OPTIMIZATION

In the proposed DEA-RNN, the weight and biases along with the size of the population are considered as the parameters to be optimized. The weight and the corresponding bias for the Elman RNN are computed using the weight matrices [62] as expressed in Eqs. (8) and (9), respectively.

$$W_n = U_n = \sum_{n=1}^{N} \alpha.\left(rand - \frac{1}{2}\right) \quad (8)$$

$$B_n = \sum_{n=1}^{N} \beta.\left(rand - \frac{1}{2}\right) \quad (9)$$

Here Wn denotes the N-th weight value of the weight matrix (n = 1, 2, . . ., N) and Bn denotes the bias value for the network. α and β are two constant parameters with the condition α and β < 1, while rand is a random number between (0,1). The RNN process is a sum of square errors arranged for each weight matrix in WC = [W1 n, W2 n, . . . , WN−1 n], where, WC is a total weights list matrix for the network. Therefore, the average sum of square errors is used as the fitness function. For the proposed DEA-RNN, the Elman RNN structure is formed with three layers: - the input layer, the hidden layer, and the output layer. Every layer has an individual index variable, i.e., i for input nodes, j and l for hidden nodes, and k output nodes. As Elman RNN has a feed-forward network structure, the input vector x is transmitted through the weight layer. The input layer vector function of the RNN is given as

$$net_j\,(t) = \sum_{i=1}^{n} x_i(t) W_{n(ji)} + B_{n(j)} \quad (10)$$

$$net_j(t) = \sum_{i=1}^{n} x_i(t) W_{n(ji)} + \sum_{l=1}^{m} y_l(t-1) U_{n(jl)} + B_{n(j)} \quad (11)$$

$$y_j(t) = f\left(net_j(t)\right) = f\left(\sum_{i=1}^{n} x_i(t) W_{n(ji)} + \sum_{l=1}^{m} y_l(t-1) U_{n(jl)} + B_{n(j)}\right) \quad (12)$$

Here, the number of inputs is denoted by n, the j-th bias value of the weight matrix is represented by Bn(j) and the input layer vector function is denoted by netj (t).

Similarly, in RNN, the input vector is propagated through the weight layer with an addition of the previous hidden activation $y_l(t-1)$ through another recurrent weight layer U_n and formulated as in Eq. (11). The output function of the hidden layer $y_j(t)$ is expressed as in Eq.(12).

$$net_k(t) = \sum_{j=1}^{M} y_j(t) W_{n(kj)} + B_{n(k)} \quad (13)$$

$$Y_k(t) = g\left(net_k(t)\right) = g\left(\sum_{j=1}^{M} y_j(t) W_{n(kj)} + B_{n(k)}\right) \quad (14)$$

Here, the number of hidden nodes is denoted by m, $f()$ indicates to the Network activation function of hidden layer and $y_j(t) = f(net_j(t))$ denotes the output function of the hidden layer and calculated as the hidden-activation function of the input vector. The output of the whole network is obtained at the end of the output layer, which is identified based on the hidden layer and group of output weights W.

Here, the output function for the output layer is identicated by $net_k(t)$, $g()$ denotes to the network activation function for the output layer, $Y_k(t) = g(net_k(t))$ is a predicted output function and W_{nkj} denotes the n weights of k-th output node and j-th hidden layer nodes. The error associated with the output layer is utilized to determine the sum of the square errors. Hence, the error at the output layer is computed as given in Eq. (15).

$$E = (T_k - Y_k) \quad (15)$$

where T_k is actual output, and Y_k is the predicted output. The performance index of the RNN is calculated as in Eq. (16).

$$V_F(x) = \frac{1}{2}\sum_{k=1}^{K} (T_k - Y_k)^T (T_k - Y_k) = \frac{1}{2}\sum_{k=1}^{K} E^T.E \quad (16)$$

Computing the average sum of square is based on the performance index and calculated as in Eq. (17),

$$V_\mu(x) = \frac{\sum_{j=1}^{N} V_F(x)}{P_i} \quad (17)$$

Here Pi indicates the number of dolphin populations in the i-th iteration. The performance index is denoted by VF (x),

and the average of performance is denoted by $V_\mu(x)$. At the end of each iteration in DEA, the average Sum of Square Errors (SSE) of ith iteration is computed as given in Eq (18).

$$SSE_i = \left\{V_\mu^1(x), V_\mu^2(x), \ldots, V_\mu^n(x)\right\} \quad (18)$$

DEA uses the Minimum Sum of square Error (MSE) as the best dolphin, and the mapped configuration (weights, bias and size of population) is chosen as the best RNN structure. MSE is calculated as in a given Eq. (19).

$$MSE = \frac{1}{NL}\sum_{i=1}^{NL} \left(Y_i - \hat{Y}_i\right)^2 \quad (19)$$

Here NL denotes the number of locations, Y_i and $\hat{Y}_i$ are the observed values and predicted values of the i-th location dolphin. Based on the chosen dolphin, the obtained optimal weight and bias are retrieved, and the weights and bias of all the layers will be updated with a small variation $\nabla X_i = x_i(t) - x_{i-1}(t)$. Therefore, the updated weights and bias are given as

$$W_n^{h+1} = U_n^{h+1} = W_n^h - \nabla X_i \quad (20)$$

$$B_n^{h+1} = B_n^h - \nabla X_i \quad (21)$$

Here h denotes the current layer of the DEA-RNN. Using this process, the RNN can be tuned effectively and applied for cyber-bullying tweets classification. Algorithm 1 presents the pseudo-code for DEA-RNN.

IV. EXPERIMENT AND ANALYSIS

In this section, the evaluation of DEA-RNN is performed over datasets crawled from Twitter utilizing these metrics: recall, precision, F-measure, accuracy, and specificity. The input dataset and the data annotation are described in sections III-A and III-B. Two baseline cyberbullying models based on deep learning, namely Bi-LSTM [21], RNN [21], and three baseline cyberbullying models based on machine learning models, namely, SVM [26], Multinomial Naive Bayes (MNB) [11], and R [11] are used for the comparison with the proposed DEA-RNN model. These models have been selected from state-of-the-art cyberbullying detection in social media. The same setup parameters configurations of the considered baseline models in the original papers are used. However, Python 3.7.4 and Pycharm IDE 2020.2.3 were used for the experiments. In the implementation and the experiments configurations, some required libraries were used, such as Keras, TensorFlow, NumPy, NLTK, Scikitlearn, Tweepy, etc. The experimental evaluations are carried out on a personal system with configurations, Intel Core-i5 CPU, Windows 10 and 8 GigaByte RAM. The preprocessing steps are performed as proposed in [58] using the NLTK Python package. The input dataset is divided into training and testing datasets. For the evaluation, it is also classified into three different scenarios 60:40% (Scenario 1), 70:30% (Scenario 2), and 90:10% (Scenario 3). The evaluation metrics are chosen to display the best performance of the tweet classification of each method. Each implemented method is run N = 20 times to obtain an average value of each evaluation metric, as well as 5-fold cross-validation is adopted.

Table.1. DEA-RNN Algorithm

Algorithm 1: DEA-RNN

```
Begin
  Initializes DEA population, size dimension and Elman RNN structure
  Load the training data
  While (MSE < stopping criteria)
    Pass the DEA locations as weights to the network
    Feed-forward network runs using the weights initialized with DEA
    For each solution candidate
      Compute the error utilizing Eq. (15)
      Minimize the error using adjusting network parameter by utilizing DEA
      Generate DEA next loop location (i + 1). (From random locations using Eq. (6))
      Eliminate a fraction of the worst solutions.
      Find new solutions to replace the old ones.
      Assess the fitness function to select the best configuration of RNN
      If new location (i + 1) > old location (i)
          Replace old location (i) with the new location (i + 1)
      End if
    End for
    DEA estimates weight and bias at each iteration Until the network is converged
    Update weights and bias utilizing Eq. (20) & (21)
  End While
End
```

4.1 Evaluation Metrics

This sub-section briefly highlights the evaluation metrics utilized in this study to evaluate the efficiency of DEA-RNN. The evaluation process is performed based on the following metrics: accuracy, recall, precision, F-measure, specificity and computing training time. However, each method is run (N = 20) times for all experiments to obtain an average of obtained results for each evaluation metric. These performance metrics are described in Table 2.

Table.2. Evaluation metrics

Metrics	Equations		Descriptions
Accuracy	$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$	(22)	The ratio of correctly total predicted observations to the total number of predictions
	$Average\ Accuracy = \frac{1}{N}\sum_{j=1}^{N} Accuracy_j$	(23)	The average accuracy for N runs is calculated using Eq.(23). Where N=20 and the accuracy is computed using Eq. (22)
Precision	$Precision = p = \frac{TP}{TP + FP}$	(24)	The ratio of the successfully estimated positive observation to the total predicted positive observations
	$Average\ Precision = \frac{1}{N}\sum_{j=1}^{N} Precision_j$	(25)	The average precision for N runs is computed utilizing Eq(25). Where N=20 and the precision is calculated utilizing Eq. (24)
Recall	$Recall = R = \frac{TP}{TP + FN}$	(26)	The ratio of successfully estimated true positive rate out of all observations in an actual positive class
	$Average\ Recall = \frac{1}{N}\sum_{j=1}^{N} Recall_j$	(27)	The average recall for N runs is calculated using Eq.(27). Where N=20 and the recall is computed using Eq. (26)
F-Measure	$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall}$	(28)	The weighted average of recall and precision values
	$Average\ F - Measure = \frac{1}{N}\sum_{j=1}^{N} F - Measure_j$	(29)	The average of F-Measure for N runs is computed utilizing Eq(29). Where N=20 and the F-Measure is calculated utilizing Eq. (28)
Specificity	$Specificity = \frac{TN}{TN + FP}$	(30)	The ability of the classifier to determine the true negative results (TNR) correctly.
	$Average\ Specificity = \frac{1}{N}\sum_{j=1}^{N} Specificity_j$	(31)	The average of specificity for N runs is calculated using Eq.(31). Where N=20 and the specificity is computed using Eq. (30)
Performance Improvement Rate(PIR)	$PIR = \lvert perf_M(proposed) - perf_M(existing) \rvert$	(32)	PIR defines the improvement rate of the suggested model in comparison with the existing model.

4.2 Experimental Results

This sub-section discusses the obtained experimental results of DEA-RNN classifier in comparison with some considered baseline deep learning models, namely Bi-LSTM, RNN, and other baseline machine learning models, namely MNB, RF, and SVM. The prediction results of cyberbullying are validated based on various input dataset scenarios 60:40% (Scenario 1), 70:30% (Scenario 2), and 90:10% (Scenario 3). The performance evaluation is carried out in terms of the aforesaid metrics.

The experiments were executed M = 20 times for each classifier over each dataset input scenario. Then, the average of the performance metrics is computed using equations as described in Table 2. The overall performance comparison results on various classifiers over different dataset input scenarios are illustrated in Table 3.

Table.3. Performance results of accuracy, recall, precision, F-measure, and specificity on different classifiers over various dataset input scenarios

Training and Testing Splitting Dataset	Methods	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Specificity (%)
Scenario 1 60:40 %	Bi-LSTM	79.46	79.05	75.77	77.38	81.32
	RNN	78.39	78.85	74.78	76.76	80.09
	SVM	75.42	74.48	69.76	72.04	79.85
	RF	71.14	72.96	67.38	70.06	77.23
	MNB	61.32	68.18	63.01	65.49	72.52
	DEA-RNN	82.25	81.29	76.33	80.29	82.84
Scenario 2 70:30 %	Bi-LSTM	83.45	82.88	82.78	81.34	86.54
	RNN	80.26	80.09	79.77	80.36	86.02
	SVM	77.10	76.60	77.14	76.87	85.68
	RF	75.14	78.96	77.08	78.02	85.03
	MNB	64.45	75.78	69.87	72.76	84.93
	DEA-RNN	87.14	87.02	87.11	87.08	87.10
Scenario 3 90:10 %	Bi-LSTM	88.74	87.90	87.52	87.71	89.47
	RNN	87.15	86.62	85.90	86.26	88.95
	SVM	85.21	84.25	82.72	83.48	87.26
	RF	83.45	83.87	82.49	83.17	88.33
	MNB	82.26	80.01	78.89	79.45	86.83
	DEA-RNN	90.45	89.52	88.98	89.25	90.94

4.2.1 Average Accuracy

The proposed DEA-RNN model is evaluated in terms of accuracy compared to the considered existing models by computing the average accuracy for all scenarios. As shown in Figure 3, the DEA-RNN model has obtained the highest average accuracy of 90.45% in scenarios 3, while other methods such as Bi-LSTM, RNN, SVM, MNB, and RF have got 88.74%, 87.15%, 85.21%, 82.26%, and 83.45%, respectively.

It is observed that the performance of deep learning models (Bi-LSTM and RNN) is better than machine learning models (SVM, RF, and MNB). The MNB model shows the worst performance among all the models. Similarly, DEA-RNN achieved 87.14%, with scenario 2, which is the best accuracy value compared to accuracy results 83.45%, 80.26%, 77.10%, 64.45%, and 75.14% obtained by other existing Bi-LSTM, RNN, SVM, MNB, and RF models respectively.

Also, in scenario 1, the proposed model achieved the optimum results of 82.25%, outperforming the considered existing models for the evaluation process. Bi-LSTM has got the second score among all the other models, whereas MNB has got the worst performance results. It can be concluded that the performance of the proposed model and other methods in Scenario 3 has optimal results than other scenarios in terms of accuracy, as illustrated in Fig. 3.

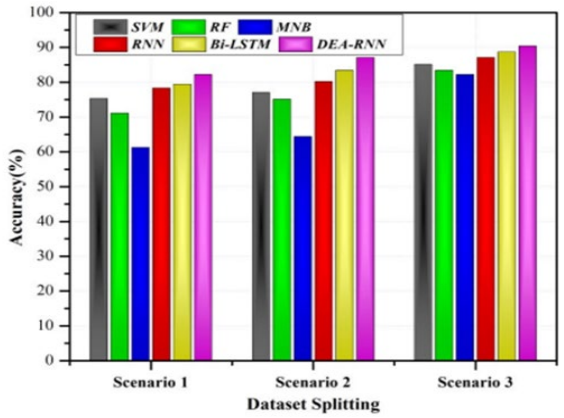

Fig.3. Performance evaluations in terms of average accuracy.

4.2.2 Average Precision

Fig. 4 shows the average precision results of the proposed DEA-RNN model compared to the considered existing models. The DEA-RNN has got 89.52 % with scenario 3, while the considered current models Bi-LSTM, RNN, SVM, MNB, and RF have obtained 87.9%, 86.62%, 84.25%, 80.01%, and 83.87%, respectively. Similarly, DEA-RNN achieved 87.02%, with scenario 2, which is the best precision value compared to precision results 82.88%, 80.09%, 76.6%, 75.78%, and 78.96% obtained by other existing Bi-LSTM, RNN, SVM, MNB, and RF models respectively.

In scenarios 2 and 3, Bi-LSTM has got the second precision score among all the other models, whereas MNB has got the worst performance results. From Fig.4, it can be clearly observed that the performance with (scenario 3) has optimal results than other scenarios in terms of precision metric.

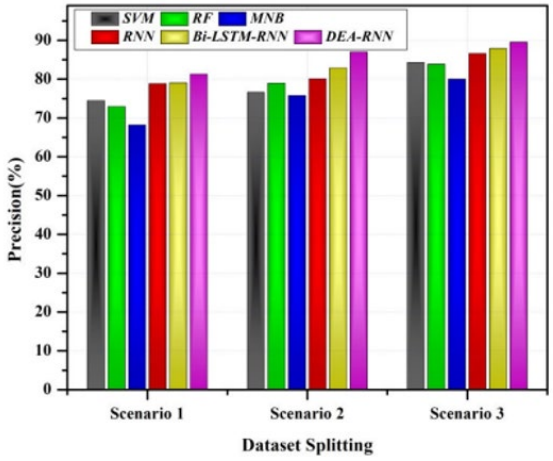

Fig.4. Performance Evaluations in Terms of Average Accuracy

4.2.3 Average Recall

The average recall of the proposed model with the compared methods is plotted in Fig 5. It can be observed that from the plot when the input dataset is scenario 3, DEA-RNN scored 88.98 %, which is the highest result among all scenarios. Besides, it is the highest result in scenario 3 compared.

4.2.4 Average F-Measure and Specificity

Fig. 6 shows the performance of the algorithms in terms of the average F-Measure (left) and the average specificity (right). DEA-RNN has got 89.25% average F-measure when the input dataset is scenario 3 (90:10 %), which is the highest result among all dataset input scenarios. While the to the recall of the considered existing models Bi-LSTM, RNN, SVM, MNB, and RF, which have obtained 87.52%, 85.9%, 82.72%, 78.89%, and 82.49%, respectively. Likewise, DEA-RNN achieved 87.11%, with scenario 2, which is the best recall value compared to recall results 82.78%, 79.77%, 77.14%, 69.87%, and 77.08% obtained by other existing Bi-LSTM, RNN, SVM, MNB, and RF models respectively. Also, in scenario 1, the suggested model got the optimum results of 76.33% outperforming the current models considered for the evaluation process. In contrast, the MNB classifier has obtained 63.01% over scenario 1, which is the lowest result. Finally, it is observed that the performance of deep learning models such as Bi-LSTM and RNN is better than machine learning models (SVM, RF, and MNB).

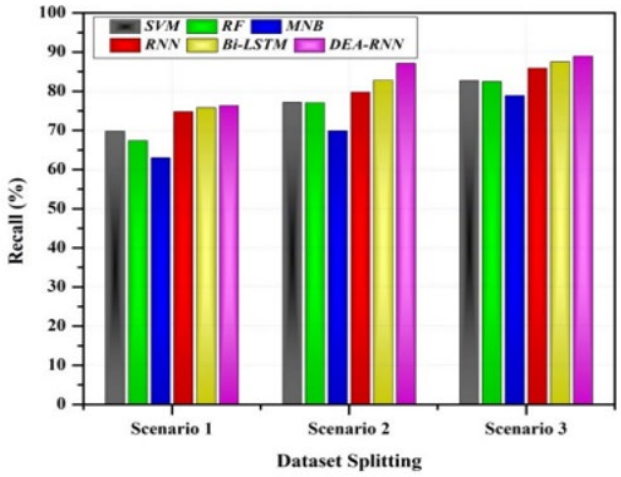

Fig.5. Performance Evaluation in Terms of Average Recall.

current models Bi-LSTM, RNN, SVM, MNB, and RF have obtained 87.71%, 86.26%, 83.48%, 79.45%, and 83.17% F-measure values, respectively. Moreover, when the input dataset scenario is 70:30%, DEA-RNN obtained 87.08%, which is the best performance compared with Bi-LSTM, RNN, SVM, MNB, and RF models. In contrast, the MNB classifier obtained 65.49% when the input splitting datasetis scenario 1, which is the lowest result. In according tothe specificity, the proposed model has achieved 90.94%of specificity in scenario 3, which is the best result com- pared with Bi-LSTM, RNN, SVM, MNB, and RF models. We can conclude that the specificity of DEA-RNN with scenario 3 has got the optimum result among all the results ofall metrics over all the scenarios, as shown in Fig. 6 (right).

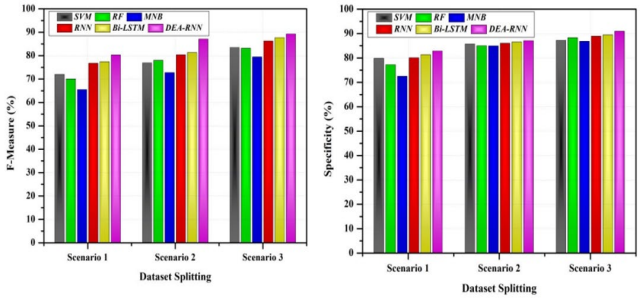

Fig.6. Performance evaluation in terms of average F-measure (left) and average specificity (right).

4.2.5 Performance Evaluation in Terms of TrainingTime

The Training time of the proposed model was compared with baseline models. Where, scenario 2 has been taken into consideration for computing the training time. It can be observed that the proposed DEA-RNN model has less training time compared to other deep learning Bi-LSTM, RNN baseline models. The training time of Bi-LSTM is more than the proposed model, RNN as well as the machine learning models, but the achievement of Bi-LSTM is better than the other baseline models and less the proposed model. DEA-RNN. has consumed 248.52 seconds in training time, whereas the baseline models based on deep learning Bi-LSTM, RNN have consumed 349.1, 274.31seconds, respectively. SVM consumed training time more than MNB and RF, But the performance of SVM model in detecting cyberbullying is more efficient than MNB and RF. We can conclude that, the other baseline models based on machine learning, suchas MNB and RF have less training time than other existing models based on deep learning including the proposed model. Figure 7 shows the Performance Improvement Rate (PIR) of the proposed DEA-RNN model compared with the con sidered current deep learning and machine learning models. The details of performance improvement have provided in

section V.

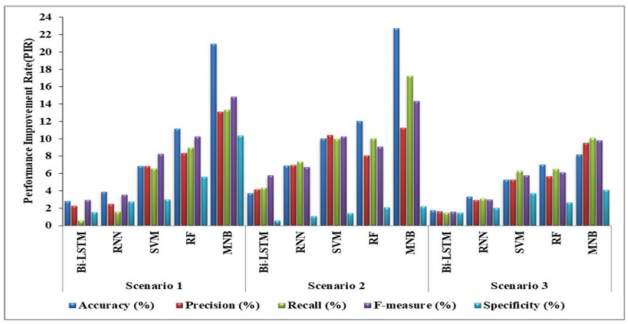

Fig.7. Performance improvement rate (PIR) of the proposed model compared to existing models.

In summary, we observe that all performance metrics(i.e., specificity, f-measure, precision, recall, and accuracy) generate the highest performance with scenario 3 than other scenarios. Also, DEA-RNN has achieved the best results for cyberbullying tweet classification in terms of all evaluation metrics on all three scenarios. In addition, DEA-RNN has attained the average of all scenarios 86.61% accuracy, 85.94% precision, 84.14% recall, 85.54% F1-score, and 86.96% specificity values which are higher than the considered state-of-the-art models. Therefore, this approach can besuggested as an effective approach for detecting CB in the Twitter. The effective solutions were attained in this model,which can be attributed to the use of DEA for the weight andbias optimization and the excellent reduction of training time. Besides, this signifies the impact of DEA on the performanceof RNN. This also ensures that the proposed DEA-RNN can be highly adaptable for modern specific short text topic models.

V. Discussion

Performance Improvement Rate (*PIR*) shows the Improvement of the suggested model in terms of the followingmetrics: specificity, f-measure, precision, recall, and accu- racy. The total *PIR* is determined by comparing the overall performance of the proposed model with the other exist-ing models, two deep learning and three machine learning models considered for the evaluation process. The improve-ment rates of the proposed model in terms of accuracy in Scenario 2 are 3.69%, 6.91%, 10.04%, 12%, and 22.69%compared with baseline models Bi-LSTM [21], RNN [21], SVM [26], RF [11], and MNB [11], respectively. Similarly, the PIR of accuracy in Scenario 3 are 1.71%, 3.3%, %,5.24%, 7%, and 8.19% compared with Bi-LSTM, RNN, SVM, RF, and MNB. In according to precision, the improvement rates of the proposed model in Scenario 2 are 4.14 %, 6.93%, 10.42%, 8.06%, and 11.24%, compared with baseline models Bi-LSTM, RNN, SVM, RF, and MNB, respectively. Like- wise, the performance improvement rate of precision in Scenario 3 is 1.62%, 2.9%, 5.27%, 5.65%, and 9.51 compared with Bi-LSTM, RNN, SVM, RF, and MNB. The improve- ment rates of the proposed model in terms of recall in Scenario 2 are 4.33%, 7.34%, 10.03%, 17.24%, 6.48%, and 10.09% compared with Bi-LSTM, RNN, SVM, RF, and MNB, respectively. Similarly, the performance improvementrate of accuracy in Scenario 3 are 1.46%, 3.08%, 6.26%, 6.48%, and 10.09% compared with Bi-LSTM, RNN, SVM, RF, and MNB. In according to F-Measure, the improvementrates of the proposed model in Scenario 2 are 5.74%, 6.72%,10.21%, 9.06%, and 14.32% compared with Bi-LSTM, RNN, SVM, RF, and MNB, respectively. Likewise, the performance improvement rate of precision in Scenario 3 are 1.54%, 2.99%, 5.77 %, 6.08%, and 9.8%, compared with Bi-LSTM, RNN, SVM, RF, and MNB.

Figure 7 shows the performance improvement rate of the proposed model compared to existing models. In brief, the overall average performance improvement rate (PIR) gained by the developed model reached 2.42%, 3.822 compared to the deep learning models Bi-LSTM and RNN, respectively. Likely, the overall average *PIR* obtained by the developed model reached 6.65%, 7.55%, and 12.12% compared to the Machine learning models SVM, RF, and MNB, respectively. Therefore, the overall improvement rates of the proposed model proves that the proposed hybrid DEA-RNN model can be suggested as an effective approach for detecting cyberbullying in the Twitter dataset. Also, DEA-RNN has achieved the best results for cyberbullying tweet classification in terms of all evaluation metrics on all three scenarios. The effective solutions were attained in this model, which can be attributed to the use of DEA for the weight and bias optimization and the excellent reduction of training time. Besides, this signifies the impact of DEA on the performance of RNN. This also ensures that the proposed DEA-RNN can be highly adaptable for modern specific short text topic models.

VI. Conclusion

This paper developed an efficient tweet classification model to enhance the effectiveness of topic models for the detection of cyber-bullying events. DEA-RNN was developed by combinding both the DEA optimization and the Elman type RNN

for efficient parameter tuning. Furthermore, it was tested in comparison with the existing Bi-LSTM, RNN, SVM,RF, and MNB methods on a newly created Twitter dataset, which was extracted using CB keywords. The experimental analysis showed that the DEA-RNN had achieved optimal results compared to the other existing methods in allthe scenarios with various metrics such as accuracy, recall, F-measure, precision, and specificity. This signifies the impact of DEA on the performance of RNN. Although the hybrid proposed model obtained higher performance rates than the other considered existing models, the feature compatibility of DEA-RNN reduces when the input data is increased greater than the initial input. The current study was limited only to the Twitter dataset exclusively; other Social Media Platforms (SMP) such as Instagram, Flickr, YouTube, Facebook, etc., should be investigated in order to detect the trend of cyberbullying. Then, the possibility of utilizing multiple source data for cyber-bullying detection will be investigated in the future. Furthermore, we performedthe analysis only on the content of tweets; we could not perform the analysis in relation to the users' behavior. This will be in future works. The proposed model works to detectcyberbullying utilizing textual content of tweets, whereas the other type of media such as images, video, and audio is still an open research area and future research directions. Besides, we aim to classify and detect CB tweets in a real-time stream.

REFERENCES

[1]. F. Mishna, M. Khoury-Kassabri, T. Gadalla, and J. Daciuk, ''Risk factors for involvement in cyber bullying: Victims, bullies and bully–victims,'' Children Youth Services Rev., vol. 34, no. 1, pp. 63–70, Jan. 2012.

[2]. K. Miller, ''Cyberbullying and its consequences: How cyberbullying is contorting the minds of victims and bullies alike, and the law's limited available redress,'' Southern California Interdiscipl. Law J., vol. 26, no. 2, p. 379, 2016.

[3]. A. M. Vivolo-Kantor, B. N. Martell, K. M. Holland, and R. Westby, ''A systematic review and content analysis of bullying and cyber-bullying measurement strategies,'' Aggression Violent Behav., vol. 19, no. 4, pp. 423–434, Jul. 2014.

[4]. H. Sampasa-Kanyinga, P. Roumeliotis, and H. Xu, ''Associations between cyberbullying and school bullying victimization and suicidal ideation, plans and attempts among Canadian schoolchildren,'' PLoS ONE, vol. 9, no. 7, Jul. 2014, Art. no. e102145.

[5]. M. Dadvar, D. Trieschnigg, R. Ordelman, and F. de Jong, ''Improving cyberbullying detection with user context,'' in Proc. Eur. Conf. Inf. Retr., in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 7814, 2013, pp. 693–696.

[6]. A. S. Srinath, H. Johnson, G. G. Dagher, and M. Long, ''Bul- lyNet: Unmasking cyberbullies on social networks,'' IEEE Trans. Computat. Social Syst., vol. 8, no. 2, pp. 332–344, Apr. 2021.

[7]. A. Agarwal, A. S. Chivukula, M. H. Bhuyan, T. Jan, B. Narayan, and M. Prasad, ''Identification and classification of cyberbullying posts: A recurrent neural network approach using under-sampling and class weighting,'' in Neural Information Processing (Communications in Computer and Information Science), vol. 1333, H. Yang, K. Pasupa, A. C.-S. Leung, J. T. Kwok, J. H. Chan, and I. King, Eds. Cham, Switzerland: Springer, 2020, pp. 113–120.

[8]. Z. L. Chia, M. Ptaszynski, F. Masui, G. Leliwa, and M. Wroczynski, ''Machine learning and feature engineering-based study into sarcasm and irony classification with application to cyberbullying detection,'' Inf. Process. Manage., vol. 58, no. 4, Jul. 2021.

[9]. N. Yuvaraj, K. Srihari, G. Dhiman, K. Somasundaram, A. Sharma, S. Rajeskannan, M. Soni, G. S. Gaba, M. A. AlZain, and M. Masud, ''Nature-inspired-based approach for automated cyberbullying classifica- tion on multimedia social networking,'' Math. Problems Eng., vol. 2021.

[10]. B. A. Talpur and D. O'Sullivan, ''Multi-class imbalance in text classification: A feature engineering approach to detect cyberbullying in Twitter,'' Informatics, vol. 7, no. 4, p. 52, Nov. 2020.

[11]. A. Muneer and S. M. Fati, ''A comparative analysis of machine learning techniques for cyberbullying detection on Twitter,'' Futur. Internet, vol. 12, no. 11, pp. 1–21, 2020.

[12]. R. R. Dalvi, S. B. Chavan, and A. Halbe, ''Detecting a Twitter cyberbullying using machine learning,'' Ann. Romanian Soc. Cell Biol., vol. 25, no. 4, pp. 16307–16315, 2021.

[13]. R. Zhao, A. Zhou, and K. Mao, ''Automatic detection of cyberbullying on social networks based on bullying features,'' in Proc. 17th Int. Conf. Dis- trib. Comput. Netw., Jan. 2016, pp. 1–6.

[14]. L. Cheng, J. Li, Y. N. Silva, D. L. Hall, and H. Liu, ''XBully: Cyberbullying detection within a multi-modal context,'' in Proc. 12th ACM Int. Conf. Web Search Data Mining, Jan. 2019, pp. 339–347.

[15]. K. Reynolds, A. Kontostathis, and L. Edwards, ''Using machine learning to detect cyberbullying,'' in Proc. 10th Int. Conf. Mach. Learn. Appl. Workshops (ICMLA), vol. 2, Dec. 2011, pp. 241–244.

[16]. S. Agrawal and A. Awekar, ''Deep learning for detecting cyberbullying across multiple social media platforms,'' in Advances in Infor- mation Retrieval (Lecture Notes in Computer Science), G. Pasi, B. Piwowarski, L. Azzopardi,

and A. Hanbury, Eds. Cham, Switzerland: Springer, 2018, pp. 141–153.

[17]. R. I. Rafiq, H. Hosseinmardi, R. Han, Q. Lv, S. Mishra, and S. A. Mattson, ‘‘Careful what you share in six seconds: Detecting cyberbullying instances in vine,’’ in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM), Aug. 2015, pp. 617–622.

[18]. N. Yuvaraj, V. Chang, B. Gobinathan, A. Pinagapani, S. Kannan,

[19]. G. Dhiman, and A. R. Rajan, ‘‘Automatic detection of cyberbullying using multi-feature based artificial intelligence with deep decision tree classi- fication,’’ Comput. Electr. Eng., vol. 92, Jun. 2021.

[20]. A. Al-Hassan and H. Al-Dossari, ‘‘Detection of hate speech in Arabic tweets using deep learning,’’ Multimedia Syst., Jan. 2021.

[21]. Y. Fang, S. Yang, B. Zhao, and C. Huang, ‘‘Cyberbullying detection in social networks using bi-GRU with self-attention mechanism,’’ Informa- tion, vol. 12, no. 4, p. 171, Apr. 2021.

[22]. C. Iwendi, G. Srivastava, S. Khan, and P. K. R. Maddikunta, ‘‘Cyberbullying detection solutions based on deep learning architectures,’’ Multimedia Syst., 2020.

[23]. B. A. H. Murshed, H. D. E. Al-ariki, and S. Mallappa, ‘‘Semantic analysis techniques using Twitter datasets on big data? Comparative analysis study,’’ Comput. Syst. Sci. Eng., vol. 35, no. 6, pp. 495–512, 2020.

[24]. P. Galán-García, J. G. De La Puerta, C. L. Gómez, I. Santos, and P. G. Bringas, ‘‘Supervised machine learning for the detection of troll pro- files in Twitter social network: Application to a real case of cyberbullying,’’ Logic J. IGPL. vol. 24, no. 1, pp. 42–53, 2015.

[25]. Y. Zhang and A. Ramesh, ‘‘Fine-grained analysis of cyberbullying using weakly-supervised topic models,’’ in Proc. IEEE 5th Int. Conf. Data Sci. Adv. Anal. (DSAA), Oct. 2018, pp. 504–513.

[26]. Z. Chen, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, ‘‘Leveraging multi-domain prior knowledge in topic models,’’ in Proc. 23rd Int. Jt. Conf. Artif. Intell. Int. Jt. Conf. Artif. Intell. (IJCAI), vol. 13, 2013, pp. 2071–2077.

[27]. N. M. G. D. Purnamasari, M. A. Fauzi, Indriati, and L. S. Dewi, ‘‘Cyberbullying identification in Twitter using support vector machine and information gain-based feature selection,’’ Indones. J. Electr. Eng. Comput. Sci., vol. 18, no. 3, pp. 1494–1500, 2020.

[28]. R. R. Dalvi, S. B. Chavan, and A. Halbe, ‘‘Detecting a Twitter cyber- bullying using machine learning,’’ in Proc. 4th Int. Conf. Intell. Com- put. Control Syst. (ICICCS), May 2020, pp. 297–301.

[29]. M. A. Al-Garadi, K. D. Varathan, and S. D. Ravana, ‘‘Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network,’’ Comput. Hum. Behav., vol. 63, pp. 433–443, Oct. 2016.

[30]. Q. Huang, V. K. Singh, and P. K. Atrey, ‘‘Cyber bullying detection using social and textual analysis,’’ in Proc. 3rd Int. Workshop Socially-Aware Multimedia (SAM), 2014, pp. 3–6.

[31]. A. Squicciarini, S. Rajtmajer, Y. Liu, and C. Griffin, ‘‘Identification and characterization of cyberbullying dynamics in an online social network,’’ in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining, Aug. 2015, pp. 280–285.

[32]. V. Balakrishnan, S. Khan, and H. R. Arabnia, ‘‘Improving cyberbullying detection using Twitter users’ psychological features and machine learning,’’ Comput. Secur., vol. 90, Mar. 2020.

[33]. K. S. Alam, S. Bhowmik, and P. R. K. Prosun, ‘‘Cyberbullying detection: An ensemble-based machine learning approach,’’ in Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mobile Netw. (ICICV), Feb. 2021, pp. 710–715.

[34]. S. Pericherla and E. Ilavarasan, ‘‘Transformer network-based word embeddings approach for autonomous cyberbullying detection,’’ Int. J. Intell. Unmanned Syst., May 2021.

[35]. V. Nahar, S. Al-Maskari, X. Li, and C. Pang, ‘‘Semi-supervised learning for cyberbullying detection in social networks,’’ in Databases Theory and Application (Lecture Notes in Computer Science), vol. 8506. Cham, Switzerland: Springer, 2014, pp. 160–171.

[36]. J.-M. Xu, K. S. Jun, X. Zhu, and A. Bellmore, ‘‘Learning from bullying traces in social media,’’ in Proc. Conf. North Amer. Chapter Assoc. Com- put. Linguistics, Hum. Lang. Technol., 2012, pp. 656–666.

[37]. L. Cheng, J. Li, Y. Silva, D. Hall, and H. Liu, ‘‘PI-bully: Personalized cyberbullying detection with peer influence,’’ in Proc. 28th Int. Joint Conf. Artif. Intell., Aug. 2019, pp. 5829–5835.

[38]. J. Chen, S. Yan, and K.-C. Wong, ‘‘Verbal aggression detection on Twitter comments: Convolutional neural network for short-text sentiment analysis,’’ Neural Comput. Appl., vol. 32, no. 15, pp. 10809–10818, Aug. 2020.

[39]. R. Zhao and K. Mao, ‘‘Cyberbullying detection based on semantic enhanced marginalized denoising auto-encoder,’’ IEEE Trans. Affect. Comput., vol. 8, no. 3, pp. 328–339, Jul. 2017.

[40]. Z. Zhang, D. Robinson, and J. Tepper, ‘‘Detecting hate speech on Twitter using a convolution-GRU based deep neural network,’’ in Proc. Eur. Semantic Web Conf. (ESWC), in Lecture Notes in Computer Science, vol. 10843.

[41]. A. GangemiAnna, A. L. Gentile, A. G. Nuzzolese, S. Rudolph, M. Maleshkova, H. Paulheim, J. Z. Pan, and M.

Alam, Eds. Cham, Switzerland: Springer, 2018, pp. 745–760.

[42]. S. Paul and S. Saha, ''Cyber BERT: BERT for cyberbullying identification,'' Multimedia Syst., no. 0123456789, Nov. 2020.

[43]. M. P. Akhter, Z. Jiangbin, I. R. Naqvi, and M. AbdelMajeed, ''Abusive language detection from social media comments using conventional machine learning and deep learning approaches,'' Multimedia Syst., Jun. 2021.

[44]. X. Zhang, J. Tong, N. Vishwamitra, E. Whittaker, J. P. Mazer, R. Kowalski, H. Hu, F. Luo, J. Macbeth, and E. Dillon, ''Cyberbullying detection with a pronunciation based convolutional neural network,'' in Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA), Dec. 2016, pp. 740–745.

[45]. H. Rosa, D. Matos, R. Ribeiro, L. Coheur, and J. P. Carvalho, ''A 'deepe,' look at detecting cyberbullying in social networks,'' in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2018, pp. 1–8.

[46]. M. A. Al-Ajlan and M. Ykhlef, ''Optimized Twitter cyberbullying detection based on deep learning,'' in Proc. 21st Saudi Comput. Soc. Nat. Com- put. Conf. (NCC), Apr. 2018, pp. 1–5.

[47]. Q. Huang, D. Inkpen, J. Zhang, and D. Van Bruwaene, ''Cyberbullying intervention based on convolutional neural networks,'' in Proc. 1st Workshop Trolling, Aggression Cyberbullying (TRAC), 2018, pp. 42–51.

[48]. V. Banerjee, J. Telavane, P. Gaikwad, and P. Vartak, ''Detection of cyberbullying using deep neural network,'' in Proc. 5th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS), Mar. 2019, pp. 604–607.

[49]. J. K. Tripathy, S. S. Chakkaravarthy, S. C. Satapathy, M. Sahoo, and V. Vaidehi, ''ALBERT-based fine-tuning model for cyberbullying analy- sis,'' Multimedia Syst., Sep. 2020.

[50]. G. K. Pitsilis, H. Ramampiaro, and H. Langseth, ''Effective hate-speech detection in Twitter data using recurrent neural networks,'' Appl. Intell., vol. 48, no. 12, pp. 4730–4742, Dec. 2018.

[51]. O. Edo-Osagie, I. Lake, O. Edeghere, and B. De La Iglesia, ''Attention-based recurrent neural networks (RNNs) for short text classification: An application in public health monitoring,'' in Proc. 15th Int. Work-Conf. Artif. Neural Netw., in Lecture Notes in Computer Science, vol. 11506. Cham, Switzerland: Springer, 2019, pp. 895–911.

[52]. M. Khodabakhsh, M. Kahani, and E. Bagheri, ''Predicting future personal life events on Twitter via recurrent neural networks,'' J. Intell. Inf. Syst., vol. 54, no. 1, pp. 101–127, Feb. 2020.

[53]. A. Kumar and N. Sachdeva, ''A Bi-GRU with attention and CapsNet hybrid model for cyberbullying detection on social media,'' World Wide Web, Jul. 2021.

[54]. L. Cheng, R. Guo, Y. Silva, D. Hall, and H. Liu, ''Hierarchical attention networks for cyberbullying detection on the Instagram social network,'' in Proc. SIAM Int. Conf. Data Mining. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2019, pp. 235–243.

[55]. J. Eronen, M. Ptaszynski, F. Masui, A. Smywiński-Pohl, G. Leliwa, and M. Wroczynski, ''Improving classifier training efficiency for automatic cyberbullying detection with feature density,'' Inf. Process. Manage., vol. 58, no. 5, Sep. 2021.

[56]. S. Paul, S. Saha, and M. Hasanuzzaman, ''Identification of cyberbullying: A deep learning based multimodal approach,'' Multimedia Tools Appl., Sep. 2020.

[57]. K. Kumari and J. P. Singh, ''Identification of cyberbullying on multimodal social media posts using genetic algorithm,'' Trans. Emerg. Telecommun. Technol., vol. 32, no. 2, pp. 1–13, Feb. 2021.

[58]. K. Kumari, J. P. Singh, Y. K. Dwivedi, and N. P. Rana, ''Bilingual cyber- aggression detection on social media using LSTM autoencoder,'' Soft Comput., vol. 25, no. 14, pp. 8999–9012, Jul. 2021.

[59]. P. Nand, R. Perera, and A. Kasture, '''How bullying is this message': A psychometric thermometer for bullying,'' in Proc. 26th Int. Conf. Com- put. Linguistics, Tech. Papers (COLING), 2016, pp. 695–706.

[60]. B. A. H. Murshed, S. Mallappa, O. A. M. Ghaleb, and H. D. E. Al-ariki, ''Efficient Twitter data cleansing model for data analysis of the pandemic tweets,'' in Studies in Systems, Decision and Control, vol. 348. Springer, 2021, pp. 93–114.

[61]. T. Anuprathibha and C. S. Kanimozhiselvi, ''Penguin search optimization-based feature selection for automated opinion mining,'' Int. J. Recent Technol. Eng., vol. 8, no. 3, pp. 648–653, 2019.

[62]. H. M. Abdulwahab, S. Ajitha, and M. A. N. Saif, ''Feature selection techniques in the context of big data: Taxonomy and analysis,'' Appl. Intell., Jan. 2022.

[63]. T. Anuprathibha and C. S. Kanimozhiselvi, ''Enhanced medical tweet opinion mining using improved dolphin echolocation algorithm-based feature selection,'' Int. J. Innov. Technol. Explor. Eng., vol. 8, no. 10, pp. 2049–2055, 2019.

[64]. D. P. Mandic and J. A. Chambers, Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability. Hoboken, NJ, USA: Wiley, 2001.