An Intelligent Metaheuristic Optimization with Deep Convolutional Recurrent Neural Network Enabled Sarcasm Detection and Classification Model

K. Kavitha[1]
Research Scholar
Department of Computer Science and Engg
Acharya Nagarjuna University, Guntur
Assistant Professor, AITAM, Tekkali

Suneetha Chittineni[2]
Associate Professor
Department of Computer Applications
RVR & JC College of Engineering
Guntur, India

***Abstract*—Sarcasm is a state of speech in which the speaker says something that is externally unfriendly with a purpose of abusing/deriding the listener and/or a third person. Since sarcasm detection is mainly based on the context of utterances or sentences, it is hard to design a model to proficiently detect sarcasm in the domain of natural language processing (NLP). Despite the fact that various methods for detecting sarcasm have been created utilizing statistical machine learning and rule-based approaches, they are unable of discerning figurative meanings of words. The models developed using deep learning approaches have shown superior performance for sarcasm detection over traditional approaches. With this motivation, this paper develops novel deep learning (DL) enabled sarcasm detection and classification (DLE-SDC) model. The DLE-SDC technique primarily involves pre-processing stage which encompasses single character removal, multispaces removal, URL removal, stop word removal, and tokenization. Next to data preprocessing, the preprocessed data is converted into the feature vector by Glove Embeddings technique. Followed by, convolutional neural network with recurrent neural network (CNN-RNN) technique is utilized to detect and classify sarcasm. In order to boost the detection outcomes of the CNN+RNN technique, a hyper parameter tuning process utilizing teaching and learning based optimization (TLBO) algorithm is employed in such a way that the classification performance gets increased. The DLE-SDC model is validated using the benchmark dataset and the performance is examined interms of precision, recall, accuracy, and F1-score.**

Keywords*—*Sarcasm detection; data classification; deep learning; feature extraction; TLBO algorithm; parameter optimization

I. Introduction

Sarcasm detection in discussions has become ever more popular amongst natural language processing (NLP) scientists with the greater usage of communicative threats on social networking platforms. Natural language is an essential data source of human emotions. Automatic sarcasm detection is repeatedly defined as an NLP problem since it mainly needs to understand the human emotions language, expressions expressed by the non-textual/textual content. Sarcasm detection has gained more attention in previous decades since it facilitates precise analysis in online reviews and comments [1]. As an illustrative approach, sarcasm utilizes word in a manner which differs from the traditional meaning and order as result of misleading polarity classification. The results obtained in this development can be used for information categorization.

Sarcasm could be deliberated as an implied form of emotion. Usually, it transmits the reverse of what has been aimed. Generally, Sarcasm is related to literary devices like satire and wit/irony i.e., utilized for insult, refutes, amuse or make fun of. Specifically, the teacher exclaimed “Credit to your hard work. I have been never impressed more in my lifetime. Lol!” these sentences might expose i.e., gratitude. But, the expression of a speaker and context demonstrate the sarcastic manner of these expressions. In the lack of visible expression, defining sarcasm in Twitter is a challenging one. A stimulating perception of sarcasm has been proposed by [2] in which the analyses were carried out in 2 sarcastic states: all centric and egocentric. The previous terms indicate that the sarcasm was observed/felt only from the participant's point of view and not from addressees’ perception and the last one indicates sarcasm being observed from the addressee and participant perspectives. The generic understanding of the result transmits the prosodic feature, the one including pattern of sounds and stress is more useful in identifying sarcasm when compared to contextual features.

Fundamental analyses of sentiment from the text mightn’t be effective for understanding the clear stimulation because of the existence of different literary devices like irony, sarcasm, and so on [3]. Thus, sarcasm detection is highly required for avoiding all kinds of misinterpretation in all kinds of transmission and for ensuring that meaning aimed in the statement is assumed accordingly. Automatically identifying sarcasm could be a difficult task that could be demonstrated by automatic sarcasm analysis and detection. Identifying sarcastic statements becomes an essential process in social networking applications since it effects the organization that mines social networking data. In spite of the existence of several potential features are extracted from text, they could be gathered into major classes, such as contextual, lexical, pragmatic, and hyperbolic features [4]. The fundamental objective of this study is to classify sarcasm into different kinds that aid in understanding the intent to hurt or level of hurt i.e., existing in the sarcastic statements. Because sarcasm may elicit a broad range of feelings in a person, it can either make the receiver laugh or, in the worst-case it might elicit a deeper sense of emotional harm. The applications of type detection might be effective in understanding the sentiments behindhand sarcasm, which offer a perspective to the sentimental condition of the person engaging in a sarcastic discussion, namely, the one on whom sarcasm was meant and the person who employs sarcasm.

Several machine learning, rule-based, deep learning, and statistical based methods have been stated in related works on automated sarcasm detection in one sentence i.e., frequently based on the content of words in isolation. This involves a variety of methods like multimodal (text image) content [5] sense disambiguation and polarity flip detection in text [6]. Previous research on detecting sarcasm in text includes pragmatic (context) and lexical (content) clues [7] such as sentiments, interjections, and punctuation alterations, which are major indicators of sarcasm [8]. The characteristics in this study are handmade and cannot be generalized due to the presence of metaphorical slang and informal language, which are often used in online communication. Current research [9, 10] use NN for learning contextual and lexical features, eliminating the necessity for handmade features with the development DL method. In this paper, word embeddings are used to train recurrent, deep convolutional, or attention-based neural networks to achieve advanced results on a variety of large-scale datasets.

This paper develops novel deep learning (DL) enabled sarcasm detection and classification (DLE-SDC) model. The DLE-SDC technique primarily involves pre-processing stage which takes place at different levels. Then, Glove Embedding technique is used for the representation of word vectors. Moreover, convolutional neural network with recurrent neural network (CNN-RNN) technique is utilized to detect and classify sarcasm. In order to boost the detection outcomes of the CNN+RNN technique, a hyper parameter tuning process using teaching and learning based optimization (TLBO) algorithm is employed in such a way that the classification performance gets increased. A wide range of simulations take place on benchmark datasets and validate the results interms of different measures.

II. LITERATURE REVIEW

In Nayel et al. [11], a method that relied on a supervised ML approach named SVM was utilized for detecting sarcasm. The presented method was calculated by an ArSarcasm-v2 dataset. The efficiency of the presented method was related to another method provided to sarcasm detection shared task and sentiment analyses. Kumar and Harish [12] proposed a new method for classifying sarcastic text with content based FS technique. The projected method is composed of 2 phase FS methods for selecting better representation features. In initial phase, traditional FS approaches like MI, IG and Chi-square are utilized for selecting appropriate features subset. The selected feature subset is additionally developed by the next phase. In following phase, k-means clustering process is utilized for selecting better representation features between same features. The selected features are categorized by 2 SVM and RF classifiers. Chatterjee et al. [13] designed features to detect sarcasm by realistic features which considered the context of word. The method is depending upon a linguistic method which defines how human differentiate among various kinds of untruth. Later, they train different ML based classifiers and relate their accuracy.

Razali et al. [14] focus on detecting sarcasm in tweets by combining DL derived features with contextual constructed feature sets. A feature set is retrieved from a CNN framework and carefully combined with the handmade feature set. Those custom feature sets are developed based on their contextual explanation. Every feature set is specifically designed for the solitary task of detecting sarcasm. The aim is to find the optimum features. Few sets are beneficial for working even if it is utilized individually. Other sets aren't really substantial without integration. The result of the experiment shows positive based on Precision, Accuracy, F1-measure, and Recall. The integration of features is categorized by ML methods for the purposes of comparison. The LR approach is considered as an optimal classification approach for this work. In Rajeswari and ShanthiBala [15], a supervised classification method viz., MNNB is utilized for detecting sarcasm, and SVM is utilized for detecting the types of sarcasm. In this work, the sarcasm is extracted from the twitters using MNNB. The tweets contain noisy messages and are managed well for efficient detection of sarcasm. Additionally, the types of sarcasm are also detected for diagnosing the state of the user.

Zhang et al. [16] proposed the utilization of NN for detecting sarcasm tweets and compared the impacts of continuous automated features using discrete manual features. Particularly, they utilize bi-directional gated RNN for capturing syntactic and semantic data on twitters, and a pooling NN for extracting contextual features manually from past twitters. Akula and Garibay [17] concentrate on identifying sarcasm in textual conversation from different societal and online platforms for networking. Eventually, they developed an interpretable DL method with gated recurrent units and multi-head self-attention. The major goal of this work [18] is the sentiment analyses of people's opinions exposed on Face book based on the present epidemic condition in lower resource language. To perform this, they have made a large scale dataset consist of 10,742 automatically categorized commentaries in the Albanian language. Moreover, in this study, they reported the effort on the development and design of sentiment analyses based on DL approach. Consequently, they reported the investigational finding attained from this presented sentiment analysis by different classification methods using static and contextualized word embedding, i.e., BERT and fast Text, validated and trained on these curate and collected datasets. Das and Kolya [19] ,the sarcastic word distribution properties of a common pop culture sarcasm corpus, which includes sarcastic speeches and dialogues, are automatically extracted. Further, they proposed an amalgamation of 4p LSTM, each contains unique activation classifier. Those models are mainly intended to effectively identifying sarcasm from the text corpus.

Sundararajan and Palanisamy [20] aim are to enhance the present methods by integrating a novel perception that categorizes the sarcasm on the basis of the levels of harshness applied. The main application of the projected study will be associating the mood of an individual to the types of sarcasm shown by him/her that can give main perceptions regarding the emotional behaviour of an individual. An ensemble-based FS approach was proposed for choosing the optimum collection of features for detecting sarcasm in tweets. This optimal collection of attributes was used to determine whether the tweets were sarcastic or not. Afterward identifying the sarcastic sentence, a multi-rule based method was projected for determining the sarcasm types. Kumar et al. [21] used Mustard, a typical conversation dataset to determine the use of an ensemble supervised learning approach for identifying sarcasm. Furthermore, it can be useful in reducing model bias and assisting decision makers in knowing how to use this model accurately. Liyuan Liu et al. [22] Proposed a method called A2Text-Net which combines auxiliary variables to improve the performance of sarcastic sentiment classification.

III. The Proposed Model

This study has developed a DLE-SDC technique to classify the presence of sarcasm. The working process is demonstrated in Fig. 1. The proposed method involves different processes namely, preprocessing, Glove based word vector representation, CNN-RNN based classification, and TLBO based parameter optimization.

A. Data Pre-processing

At the first stage, the data is pre-processed to transform into a compatible format. The different sub processes involved in data pre-processing are:

- Remove single letter words.
- Remove multiple spaces.
- Remove punctuation marks.
- Remove numbers.
- Remove stop words and.
- Convert uppercase characters into lowercase.

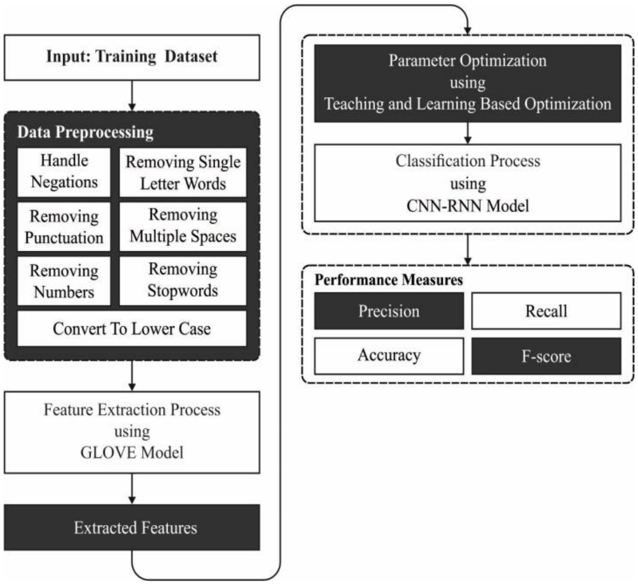


Fig. 1. Working Process of DLE-SDC Model.

B. Glove based Word Representation

The Glove approach can able to generate a vector depiction of words in the application of similarity between words as invariant. It utilizes 2 different methods as CBOW and Skip-gram. The problem related to the conventional methods includes minimum accuracy, maximum processing time, etc. The primary objective of Glove is to incorporate the approaches proposed by 2 techniques wherein optimal accuracy must be assured. In previous to generating Glove approach, the vector depiction of words has been determined. The approaches are employed to generate a vector using standard dimensions (d) for all the words. The approach that employs similarity between 2 words as invariant wherein the words in similar contents is taken into account and show same meaning.

Assume the terms beforehand presenting a formulation of Glove:

- Take a matrix of word to word co-existence count as denoted by X, whereas values X_{ij} store the amount of iterations in a word j in a sentence of word i.
- Assume that $\mathrm{X}_{\mathrm{i}} = \sum_{\mathrm{k}} \mathrm{X}_{\mathrm{ik}}$ represents the amount of times a word could be repeating in content of word i.
- Finally, consider $\mathrm{P}_{\mathrm{ij}} = \mathrm{P}(\mathrm{j}|\mathrm{i}) = \mathrm{X}_{\mathrm{ij}}/\mathrm{X}_{\mathrm{i}}$ denotes a likelihood of word j displayed in context of word i.

Let us take 2 words i & j i.e., associated with each other in content; e.g., suppose that cricket is a subject matter so that $\mathrm{i} =$ duck and $\mathrm{j} =$ boundary. Analyzing a ratio of coexistence probability with distinct probe words, k, reveals the relationships between those words. In wordsk, i.e., associated with duck by not including the boundary, let $\mathrm{k} = \mathrm{out}$, thus the ratio $\mathrm{P}_{\mathrm{ik}}/\mathrm{P}_{\mathrm{jk}}$is maximalized. Similarly, in words k depends on boundary by not including duck, let $\mathrm{k} = \mathrm{six}$, so that ratio is minimalized. Hence, words k like score i.e., appropriate to duck and boundary, as ratio is nearly 1. Conventional logic suggests that the proportion of coexistence possibilities might be used as a starting point to calculate the similarity between those terms. The ratio $\mathrm{P}_{\mathrm{ik}}\mathrm{P}_{\mathrm{jk}}$ is based on 3 words j, and k, in which standard method simulates the process as follows,

$$F\left(w_i, w_j, \widehat{w}_k\right) = \frac{P_{ik}}{P_{jk}}, \quad (1)$$

Whereas $w \in R^d$ denotes word vector and $\widehat{w} \in R^d$represents separate context word vector. As vector spaces are integrally linear structures and assume vector variances.

$$F((w_i - w_j)^T, \widehat{w}_k) = \frac{P_{ik}}{P_{jk}} \quad (2)$$

The application of algebraic function and group theory are given below:

$$F((w_i - w_j)^T \widehat{w}_k) = \frac{F(w_i^T \widehat{w}_k)}{F\left(w_j^T \widehat{w}_k\right)}, \quad (3)$$

$$where, F(w_i^T \widehat{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}$$

$$w_i^T \widehat{w}_k = log(P_{ik}) = \log{(X_{ik})} - \log{(X_i)}$$

$$\exp\,(w_i^T.\hat{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}$$

In which '.' represents a dot product between 2 vectors w_i and $w_k$whereas exp indicates an exponent function.

$$w_i^T.\hat{w}_k = log(P_{ik}) = log(X_{ik}) - log(X_i), \quad (4)$$

and the $\log(X)$ represents a constant in word i, the aforementioned operation is altered by:

$$w_i^T.\hat{w}_k + b_i + \hat{\mathrm{b}}_{\mathrm{k}} = log(X_{ik}), \quad (5)$$

In which $b_i$represents a bias for word i and $\hat{\mathrm{b}}_{\mathrm{k}}$indicates a bias for word k.

Later, the optimum function from ML perception is written as follows: RHS from previous equations are calculated from corpus, which should be updated from LHS to produce relevant RHS. Thus, hypothesis (h) determines LHS, while RHS is referred to as output (y). The cost function is then converted using the least square method:

$$J = \sum_{i,k}(\,w_i^T.\hat{w}_k + b_i + \hat{\mathrm{b}}_{\mathrm{k}} - \log(X_{ik}))^2, \quad (6)$$

and it is necessary to minimize the cost function. But, previous to employing GD, the cost function should be increase dramatically by some weights to each two words; hence, a cost function may be thought of as a memory that preserves data depending on previously calculated values.

$$J = \sum f\,(X_{ij})(w_i^T.\hat{w}_k + b_i + b_k^{\wedge} - \log(X_{ik}))^2, \quad (7)$$

whereas $f(X_{ij})$ denotes weight associated with co-existence of term i with j. Generally, f is called as:

$$(\frac{x}{x_{\max}})^{\alpha}, if x < x_{\max},$$

$$1, or\ else$$

Later, when a partial derivative of J is handled by w_i as follows:

$$\frac{\partial J}{\partial w_i} = \sum_{k=1}^{d} \frac{\partial J}{\partial w_{i,k}},$$

In which d implies a dimension of word i.

If w_i is a vector for (x_1, x_2, x_3), $\frac{\partial J}{\partial w_i}$ would be:

$$\left(\frac{\partial J}{\partial w_{i,1}} = (1,0,0)\right) + \left(\frac{\partial J}{\partial w_{i,2}} = (0,1,0)\right) + \left(\frac{\partial J}{\partial w_{i,3}} = (0,0,1)\right) \quad (8)$$

Therefore, a vector of $(1,1,1)$. Hence, the manner of calculating derivatives is based on a word wherein GD is employed by learning rate alpha $(= 0.5)$, to train $word2Vec$ module. Consequently, when the module is trained, the words with similar meanings are extracted with producing an arbitrary word as input. In businesses, words with similar meanings such as business, Market, industry, products, share market, stock, etc.

C. Sarcasm Detection using CNN-RNN Technique

The extracted feature word vectors are fed into the CNN-RNN technique for the classification of sarcasm. RNN is a kind of NN which preserves internal hidden state for modelling dynamic temporal behaviour of series using random lengths by directed cyclic relations among its unit. It could be taken into account as a hidden Markov method extension which applies nonlinear transition function and can able to model long-term temporal dependency. LSTM prolongs RNN by including a forget gate f for controlling either to forget the present state; an input gate i for indicating whether it read the input; an output gate o for controlling either to output the states [23]. That gate enables LSTM for learning long-term dependencies in a series, and also facilitates it for optimizing since that gate helps the input signal to efficiently broadcast via the recurrent hidden state $r(t)$ without influencing the output. Also, LSTM efficiently handles the gradient exploding or vanishing problems which usually appear in RNN training. Fig. 2 illustrates the framework of CNN model.

$$\begin{aligned}
x_t &= \delta\big(U_r.r(t-1) + U_w w_k(t)\big) \\
i_t &= \delta\left(U_{i_r} r(t-1) + U_{i_\omega} w_k(t)\right) \\
f_t &= \delta\left(U_{f_r} r(t-1) + U_{f_\omega} w_k(t)\right) \qquad (9) \\
o_t &= \delta\left(U_{o_r} r(t-1) + U_{o_w} w_k(t)\right) \\
r(t) &= f_t \odot r(t-1) + i_t \odot x_t \\
o(t) &= r(t) \odot o(t)
\end{aligned}$$

In which $\delta(.)$ denotes an activation function, $\odot$ indicates the product using gate value, and different W matrices are learned parameters. They use the rectified linear unit (ReLU) as the activation function in this performance. A new CNN-RNN architecture is employed for multi label classification problems. It consists of: The CNN extract semantic representation from the image; the RNN models label or image relations and label dependency. The recurrent, label and image depictions are proposed to the similar low dimension space for modelling the label redundancy and the image text relation.

Fig. 3 demonstrates the structure of RNN model. The RNN method is applied as a compact but a strong representation of the label co-existence dependencies in this space. It takes the embedding of the predictive labels at every time step and maintains a hidden state for modelling the label's co-existence data. The a priori likelihood of a label provided the previous prediction label could be calculated based on their dot product with the addition of recurrent and image embeddings.

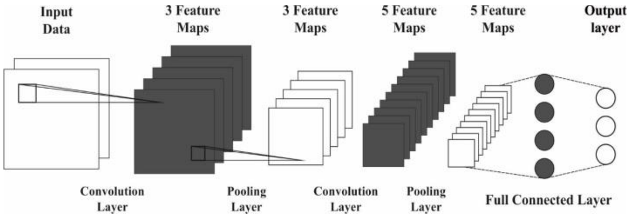


Fig. 2. Structure of CNN.

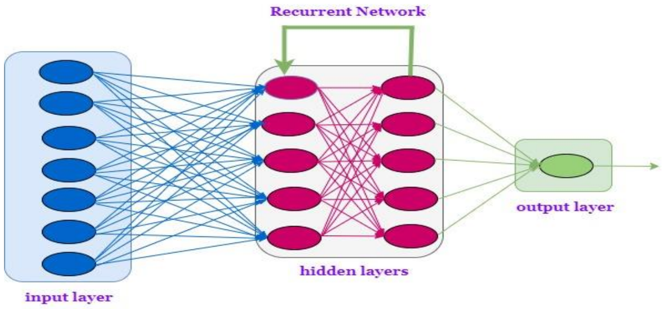


Fig. 3. Architecture of RNN.

A label k is denoted as a one-hot vector $e_k = [0, \dots 0, 1, 0, \dots, 0]$, i.e., 1 at kth position, and 0 somewhere else. The label embedding could be attained by multiplying the one-hot vector using a *label embedding matrix* U_l. The kth row of U_ι denotes the label embedding of label k.

$$w_k = U_l . e_k. \tag{10}$$

The dimension of w_k is generally lower compared to the amount of labels. The recurrent layer takes the label embedding of the previous prediction label, and models the co-existence dependency in its hidden *recurrent state* by learning nonlinear function:

$$o(t) = h_o\big(r(t-1), w_k(t)\big), r(t) = h_r\big(r(t-1), w_k(t)\big) \tag{11}$$

whereas $r(t)$ and $o(t)$ represents the hidden state and output of the recurrent layer at the time step t, correspondingly, $w_k(t)$ indicates the label embedding of t-th label in the predictive path, and $h_o(.)$, h_r $(.)$ signifies the nonlinear RNN function. The image depiction and output of recurrent layer are proposed to the similar low dimension space as the label embedding.

$$x_t = h(U_o^x o(t) + U_I^x I), \tag{12}$$

whereas U_o^x & U_I^x denotes the prediction matrix for image depiction and recurrent layer output, correspondingly. The column count of U_o^x & U_I^x indicates the similar as label embedding matrix $U_l . I$ represents the CNN image depiction. They would display in second that the learned joint embedding efficiently characterizes the significance of labels and images. Lastly, the label score could be calculated by multiplying the transpose of U_l & x_t for computing the distances among x_t and every label embedding.

$$s(t) = U_l^T x_t. \tag{13}$$

The prediction label likelihood could be calculated by soft max regularization on the scores. Fig. 4 depicts the architecture of LSTM model.

D. *Hyperparameter Optimization using TLBO Algorithm*

At the final stage, the learning rate of the CNN-RNN technique is optimally chosen by the use of TLBO algorithm in such a way that the sarcasm detection outcome gets increased. TLBO technique is a novel type of metaheuristic approach which is dependent upon teaching– learning model. It can be established by Rao et al. [24] for solving optimization issues. It can be simulated as feeding the knowledge in a class where students initially gain information from teacher and next with mutual interface. The TLBO technique has population based optimized technique where the set or class of students regarded as population. Therefore, the student of class signifies the possible solution of difficulty. The TLBO technique includes two stages as given below.

1) Teacher level: This level defines the learning of student from teacher. The teachers attempt for improving the knowledge level of student and uses for obtaining optimum marks. However, the student gains information and attain marks based on quality of teaching distributed as teacher and quality of student existing in the class. In order to simulate, supposing there are ‘n’ amount of subjects ($j = 1,2, \dots, n$) existing to ‘ N_p ’ amount of students (population size, $i = 1,2, \dots N_p$). In some teaching-learning cycles (iteration, $k = 0,1, 2, \dots I_n$), M_j^k represents the mean outcome of students in specific subject ‘j’. The teacher is one of the skilled, experienced, and extremely learned person in society. For simulating this model, an optimum student (possible solution) in total population was regarded as teacher. The variance among the outcome of teacher and the mean outcome of students in subject ‘j’ is provided as:

$$D_j^k = r\big(X_{T,j}^k - T_F M_j^k\big) \tag{14}$$

where T_F implies the teaching influence that decided the value of mean that altered and r implies the arbitrary number in range 0 to 1. T_F signifies not parameter of TLBO technique and their value is either be one or two [25]. The possible solution (student) is enhanced by moving its places near the place of an optimum possible solution (teacher) by taking into account the present mean value of possible solution. For simulating this detail, the i^{th} possible solution in the population at k^{th} teaching-learning cycle is upgraded based on subsequent written as:

$$X_{new,i,j}^k = X_{old,i,j}^k + D_j^k \tag{15}$$

When $X_{new,i}^k$ implies the superior to$X_{old,i}^k$, then $X_{new,i}^k$ is recognized; Then it can be rejected. Every accepted possible solution is continued and these developed the input to student phase.

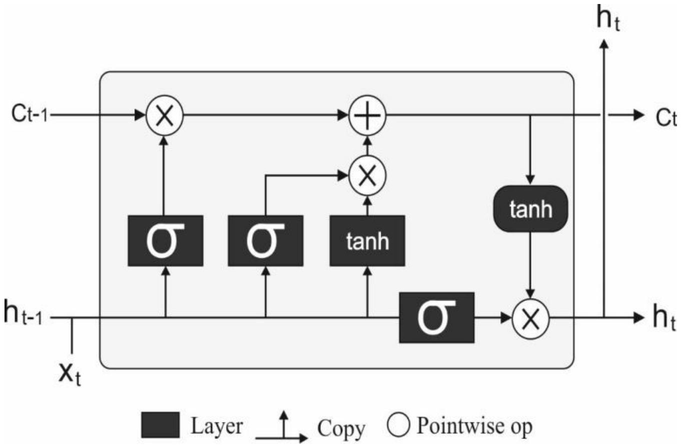


Fig. 4. Framework of LSTM.

2) Student level: Here, the student gains information with mutual communication. The students interrelate arbitrarily with another student of class for improving knowledge. Therefore, when the student (v) has superior to student (u), afterward student (u) is stimulated near student (v). Then, student (u) was stimulated away from student (v). The learning viewpoint of this phase is provided here. Two students (possible solution, X_u^k, X_v^k) are arbitrarily elected from class (population), where u, v are 2 integers arbitrary number go to [1, N_p] and $u \neq v$.

```
If $\mathrm{F}(\mathrm{X}_\mathrm{u}^\mathrm{k}) > F(\mathrm{X}_\mathrm{v}^\mathrm{k})$
        $\mathrm{X}_{\mathrm{new\_SP,u,j}}^\mathrm{k} = \mathrm{X}_{\mathrm{u,j}}^\mathrm{k} + \mathrm{r}(\mathrm{X}_{\mathrm{u,j}}^\mathrm{k} - \mathrm{X}_{\mathrm{v,j}}^\mathrm{k})$
Else
        $\mathrm{X}_{\mathrm{new\_SP,u,j}}^\mathrm{k} = \mathrm{X}_{\mathrm{u,j}}^\mathrm{k} + \mathrm{r}(\mathrm{X}_{\mathrm{v,j}}^\mathrm{k} - \mathrm{X}_{\mathrm{u,j}}^\mathrm{k})$
Endif
```

where F(x) implies the fitness function (FF) which is utilized for finding the fitness value of possible solutions, $\mathrm{X}_{\mathrm{new\_SP,u,j}}^\mathrm{k}$ refers the jth design variable of altered possible solutions from student level at kth teaching-learning iteration.

Afterward, the fitness value of $\mathrm{X}_{\mathrm{new\_SP,u}}^\mathrm{k}$ is estimated.

```
If $\mathrm{F}\left(\mathrm{X}_{\mathrm{new_{SP},u}}^\mathrm{k}\right) > F(\mathrm{X}_{\mathrm{new,u}}^\mathrm{k})$
        $\mathrm{X}_{\mathrm{new,u}}^\mathrm{k} = \mathrm{X}_{\mathrm{new_{SP},u}}^\mathrm{k}$
Else
        $\mathrm{X}_{\mathrm{new,u}}^\mathrm{k} = \mathrm{X}_{\mathrm{new,u}}^\mathrm{k}$
End if
```

IV. Performance Validation

A. Implementation Setup

The DLE-SDC technique is implemented on Python 3.6.5 tool with additional packages such as tensorflow-gpu==2.2.0, pandas, nltk, tqdm, scikit-learn, matplotlib, seaborn, pretty table, pyqt5==5.14, pycm, and numpy==1.19.5. The DLE-SDC technique is tested using a News Headlines Dataset For Sarcasm Detection dataset from Kaggle Repository [26]. The execution process of the DLE-SDC technique is given in Appendix (Fig. 13 to 16).

B. Results Analysis

This section examines the sarcasm detection performance of the DLE-SDC technique against several aspects. The DLE-SDC technique is investigated interms of different measures namely precision, recall, accuracy, and F-measures. The confusion matrix generated by the DLE-SDC technique on the classification of sarcasm is depicted in Fig. 5. The figure showcased that the DLE-SDC technique has classified a total of 14166 instances into non-sarcastic and 12639 instances into sarcastic ones.

A brief classification results analysis of the DLE-SDC technique with other DL techniques takes place in Table I and Fig. 6. From the resultant values, it is noticeable that the CNN-TLBO algorithm has offered lower classification outcomes with the precision of 0.9021, recall of 0.8946, accuracy of 0.8974, and F-measure of 0.8943. Then, the RNN-TLBO algorithm has gained somewhat increased performance with the precision of 0.9135, recall of 0.9036, accuracy of 0.9067, and F-measure of 0.9035. Eventually, the CNN-RNN model has obtained moderately closer outcome with the precision of 0.9275, recall of 0.9160, accuracy of 0.9192, and F-measure of 0.9182. However, the DLE-SDC technique has outperformed all the other DL models with the precision of 0.9406, recall of 0.9401, accuracy of 0.9405, and F-measure of 0.9403.

Accuracy analysis of the DLE-SDC technique is investigated under varying numbers of epochs in Fig. 7. The figure showcased that the training and training accuracy values get increased with an increase in epoch count. Particularly, validation accuracy is found to be superior to training accuracy.

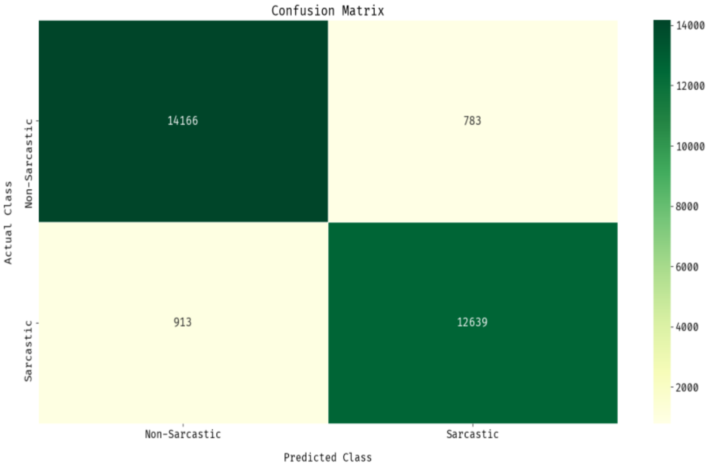


Fig. 5. Confusion Matrix of the DLE-SDC Model.

TABLE I. Results Analysis of Proposed DLE-SDC Model with Various Deep Learning Models

Methods	Precision	Recall	Accuracy	F-Measure
Proposed DLE-SDC	0.9406	0.9401	0.9405	0.9403
CNN-RNN	0.9275	0.9160	0.9192	0.9182
RNN-TLBO	0.9135	0.9036	0.9067	0.9035
CNN-TLBO	0.9021	0.8946	0.8974	0.8943

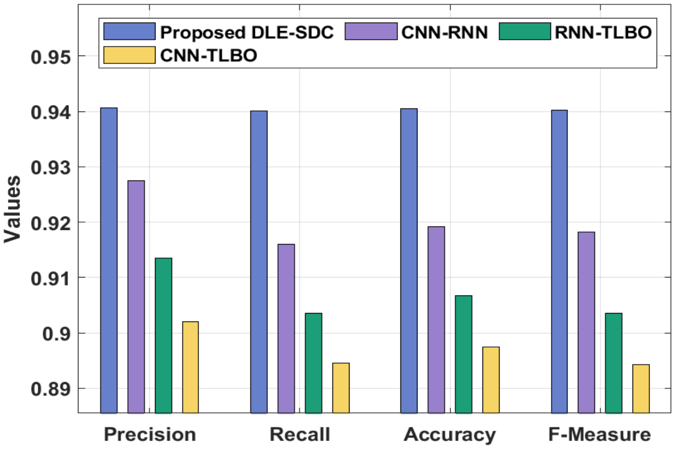


Fig. 6. Result Analysis of DLE-SDC Model with Different Measures.

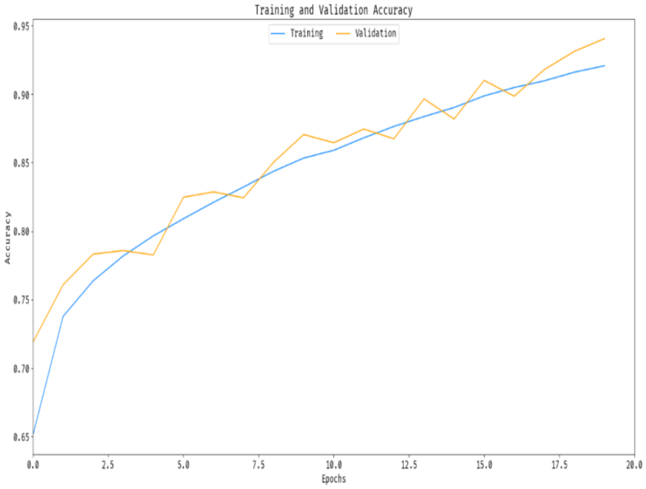


Fig. 7. Accuracy Graph on Proposed DLE-SDC Model.

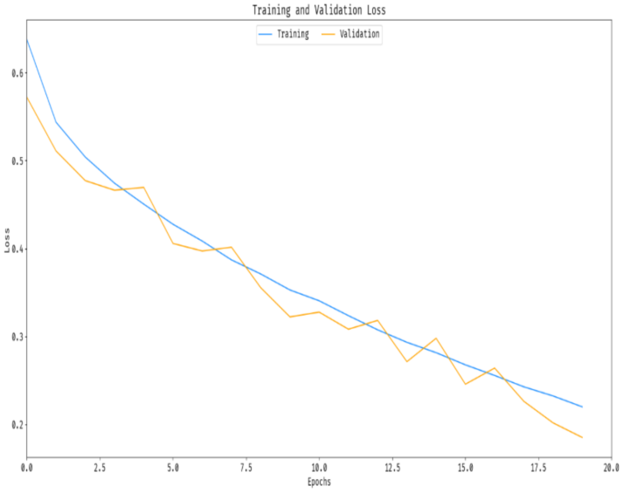


Fig. 8. Loss Graph on Proposed DLE-SDC Model.

A loss graph analysis of the DLE-SDC technique is examined under variable number of epochs in Fig. 8. The figure has shown that the training and training loss values get reduced with a rise in number of epoch. Particularly, the validation loss seems to be lower than the training loss of the DLE-SDC technique.

Fig. 9 examines the ROC analysis of the proposed DLE-SDC technique on the applied dataset. The figure portrayed that the proposed DLE-SDC technique has accomplished better performance with a maximum ROC of 0.98.

Finally, a comprehensive comparative study of the DLE-SDC technique with other techniques takes place in Table II [27]. Fig. 10 displays the precision analysis of the DLE-SDC technique with other techniques. The figure portrayed that the GRNN and VLSTM techniques have offered ineffective outcomes with the least precision of 0.663 and 0.673 respectively. Also, the E-BiLSTMA and ALSTM techniques have showcased slightly improved outcomes with the precision of 0.684 and 0.687 respectively. Moreover, the VCNN, NBOW, E-BiLSTM, and E-BiLSTMMF techniques have reached a moderately closer precision of 0.71, 0.712, 0.759, and 0.778 respectively. Furthermore, the MMNSS and A2Text-Net techniques have obtained competitive outcomes with the precision of 0.857 and 0.917. However, the proposed DLE-SDC technique has resulted in superior outcomes with a maximum precision of 0.941.

Fig. 11 showcases the recall analysis of the DLE-SDC approach with other methods. The figure demonstrated that the NBOW and GRNN approaches have offered ineffective results with the minimum recall of 0.623 and 0.647 correspondingly. In line with this, the VCNN and VLSTM manners have exhibited somewhat increased results with the recall of 0.671 and 0.672 correspondingly. Furthermore, the ALSTM, E-BiLSTMA, E-BiLSTMF, and E-BiLSTM methodologies have reached a moderately closer recall of 0.686, 0.708, 0.735, and 0.750 correspondingly. Along with that, the MMNSS and A2Text-Net methods have attained competitive results with the recall of 0.892 and 0.910. Eventually, the projected DLE-SDC approach has resulted in higher results with the maximal recall of 0.940.

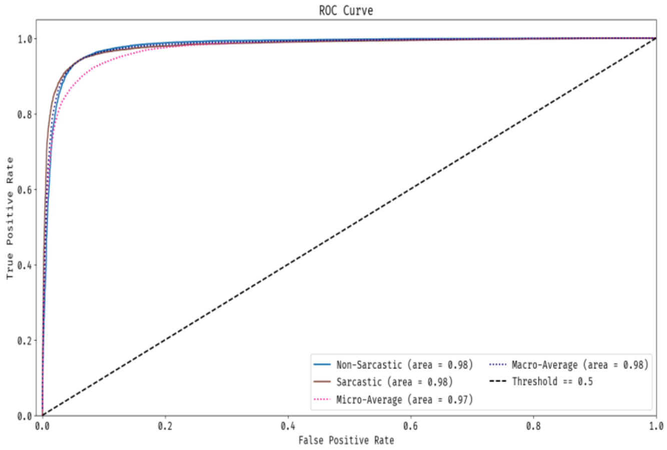


Fig. 9. ROC Analysis on Proposed DLE-SDC Model.

TABLE II. RESULTS ANALYSIS OF EXISTING WITH PROPOSED DLE-SDC MODEL IN TERMS OF DIFFERENT MEASURES

Methods	Precision	Recall	F-Measure
Proposed DLE-SDC	0.941	0.940	0.940
MMNSS	0.857	0.892	0.871
NBOW	0.712	0.623	0.641
VCNN	0.710	0.671	0.685
VLSTM	0.673	0.672	0.672
ALSTM	0.687	0.686	0.687
GRNN	0.663	0.647	0.654
E-BiLSTM	0.759	0.750	0.759
E-BiLSTMF	0.778	0.735	0.753
E-BiLSTMA	0.684	0.708	0.694
A2Text-Net	0.917	0.910	0.900

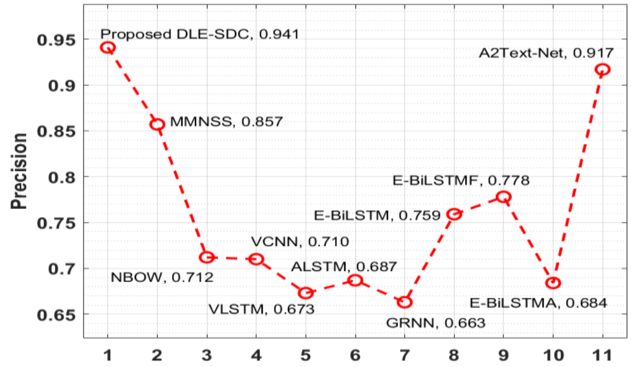


Fig. 10. Precision Analysis of DLE-SDC Model with Existing Techniques.

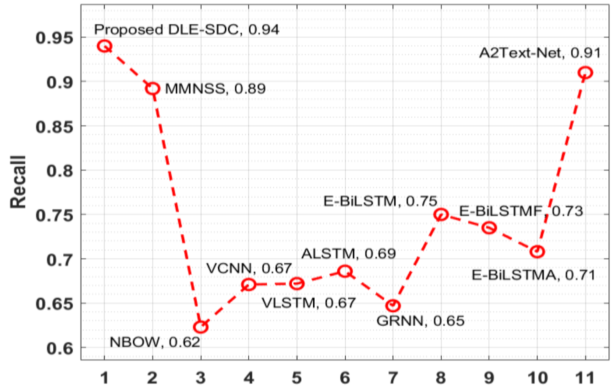


Fig. 11. Recall Analysis of DLE-SDC Model with Existing Techniques.

Fig. 12 depicts the F-measure analysis of the DLE-SDC method with other algorithms. The figure outperformed that the NBOW and GRNN methods have offered ineffective outcomes with the worse F-measure of 0.641 and 0.654 correspondingly. Similarly, the VLSTM and VCNN techniques have outperformed somewhat increased results with the F-measure of 0.672 and 0.685 correspondingly. In line with, the ALTSM, E-BiLSTMA, E-BiLSTMMF, and E-BiLSTM algorithms have reached a moderately closer F-measure of 0.687, 0.694, 0.753, and 0.759 respectively. Furthermore, the MMNSS and A2Text-Net methodologies have obtained competitive outcomes with the F-measure of 0.871 and 0.900. However, the proposed DLE-SDC approach has resulted in maximum results with the superior F-measure of 0.940.

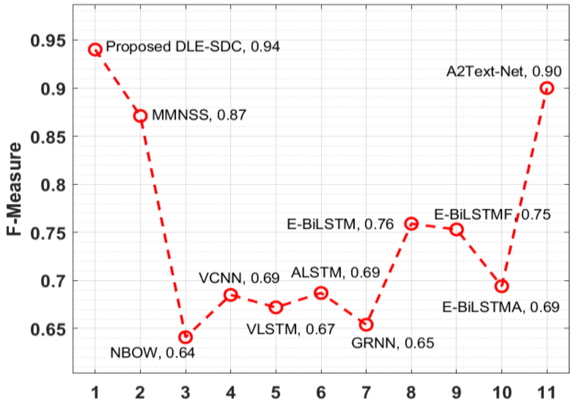


Fig. 12. F-measure Analysis of DLE-SDC Model with Existing Techniques.

From the above mentioned results analysis, it is observed that the DLE-SDC technique has accomplished maximum sarcasm detection performance and can be employed to detect sarcasm in online social media content.

V. Conclusion

This paper has presented a new DLE-SDC technique to identify and classify the sarcasm using DL technique. The proposed DLE-SDC technique comprises different stages of operations such as pre-processing, word vector representation, CNN+RNN based classification, and TLBO based hyper parameter optimization. Besides, the CNN-RNN technique involves the BiLSTM model of the detection and classification of sarcasm. In order to increase the sarcasm detection performance of the CNN-RNN model, TLBO algorithm is applied to determine the optimal learning rate of the presented CNN-RNN model and it is mainly used to boost the detection performance to a maximum extent. A wide range of simulations take place on benchmark datasets and validate the results interms of different measures. The simulation outcomes pointed out the supremacy of the DLE-SDC technique over the recent state of art techniques. As a part of future work, the sarcasm detection performance can be extended to the design of feature selection and clustering techniques.

References

[1] N. Majumder, S. Poria, H. Peng et al., “Sentiment and sarcasm classification with multitask learning,” IEEE Intelligent Systems, vol. 34, no. 3, pp. 38–43, 2019.

[2] G. Deliens, K. Antoniou, E. Clin, and M. Kissine, “Perspective-taking and frugal strategies: evidence from sarcasm detection,” Journal of Pragmatics, vol. 119, pp. 33–45, 2017.

[3] A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, and M. AbdelBasset, “Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network,” IEEE Access, vol. 7, pp. 23319–23328, 2019.

[4] S. K. Bharti and S. B. Korra, “Sarcasm detection in twitter data: a supervised approach,” in Semantic Web Science and RealWorld applications, M. D. Lytras, N. Aljohani, E. Damiani, and K. T. Chui, Eds., pp. 246–272, IGI Gobal, Hershey, PA, USA, 2019.

[5] D. Ghosh, W. Guo, and S. Muresan, “Sarcastic or not: word embeddings to predict the literal or sarcastic meaning of words,” in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1003–1012, Lisbon, Portugal, 2015.

[6] S. Castro, D. Hazarika, V. Pérez-Rosas, R. Zimmermann, R. Mihalcea, and S. Poria, “Towards multimodal sarcasm detection (an _obviously_ perfect paper),” in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4619–4629, Florence, Italy, 2019.

[7] Kreuz, R.J.; Caucci, G.M. Lexical influences on the perception of sarcasm. In Proceedings of the Workshop on Computational Approaches to Figurative Language, Association for Computational Linguistics, Rochester, NY, USA, 26 April 2007; pp. 1–4.

[8] Joshi, A.; Sharma, V.; Bhattacharyya, P. Harnessing context incongruity for sarcasm detection. In Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP, Beijing, China, 26–31 July 2015; pp. 757–762.

[9] Ghosh, A.; Veale, T. Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal. In Proceedings of the 2017 Conference on EMNLP, Copenhagen, Denmark, 7–11 September 2017; pp. 482–491.

[10] Ghosh, D.; Fabbri, A.R.; Muresan, S. Sarcasm analysis using conversation context. Comput. Linguist. 2018, 44, 755–792.

[11] Nayel, H., Amer, E., Allam, A. and Abdallah, H., 2021, April. Machine learning-based model for sentiment and sarcasm detection. In

Proceedings of the Sixth Arabic Natural Language Processing Workshop (pp. 386-389).

[12] Kumar, H.K. and Harish, B.S., 2018. Sarcasm classification: a novel approach by using content based feature selection method. Procedia computer science, 143, pp.378-386.

[13] Chatterjee, N., Aggarwal, T. and Maheshwari, R., 2020. Sarcasm detection using deep learning-based techniques. In Deep Learning-Based Approaches for Sentiment Analysis (pp. 237-258). Springer, Singapore.

[14] Razali, M.S., Halin, A.A., Ye, L., Doraisamy, S. and Norowi, N.M., 2021. Sarcasm Detection Using Deep Learning With Contextual Features. IEEE Access, 9, pp.68609-68618.

[15] Rajeswari, K. and ShanthiBala, P., 2018. Sarcasm detection using machine learning techniques. Int J Recent Sci Res., 9, pp.26368-26372.

[16] Zhang, M., Zhang, Y. and Fu, G., 2016, December. Tweet sarcasm detection using deep neural network. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: technical papers (pp. 2449-2460).

[17] Akula, R. and Garibay, I., 2021. Interpretable Multi-Head Self-Attention Architecture for Sarcasm Detection in Social Media. Entropy, 23(4), p.394.

[18] Kastrati, Z., Ahmedi, L., Kurti, A., Kadriu, F., Murtezaj, D. and Gashi, F., 2021. A Deep Learning Sentiment Analyser for Social Media Comments in Low-Resource Languages. Electronics, 10(10), p.1133.

[19] Das, S. and Kolya, A.K., 2021. Parallel Deep Learning-Driven Sarcasm Detection from Pop Culture Text and English Humor Literature. In Proceedings of Research and Applications in Artificial Intelligence (pp. 63-73). Springer, Singapore.

[20] Sundararajan, K. and Palanisamy, A., 2020. Multi-rule based ensemble feature selection model for sarcasm type detection in twitter. Computational intelligence and neuroscience, 2020.

[21] Kumar, A., Dikshit, S. and Albuquerque, V.H.C., 2021. Explainable Artificial Intelligence for Sarcasm Detection in Dialogues. Wireless Communications and Mobile Computing, 2021.

[22] Liu, Liyuan, Jennifer Lewis Priestley, Yiyun Zhou, Herman E. Ray, and Meng Han. "A2text-net: A novel deep neural network for sarcasm detection." In 2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI), pp. 118-126. IEEE, 2019.

[23] Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C. and Xu, W., 2016. Cnn-rnn: A unified framework for multi-label image classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2285-2294).

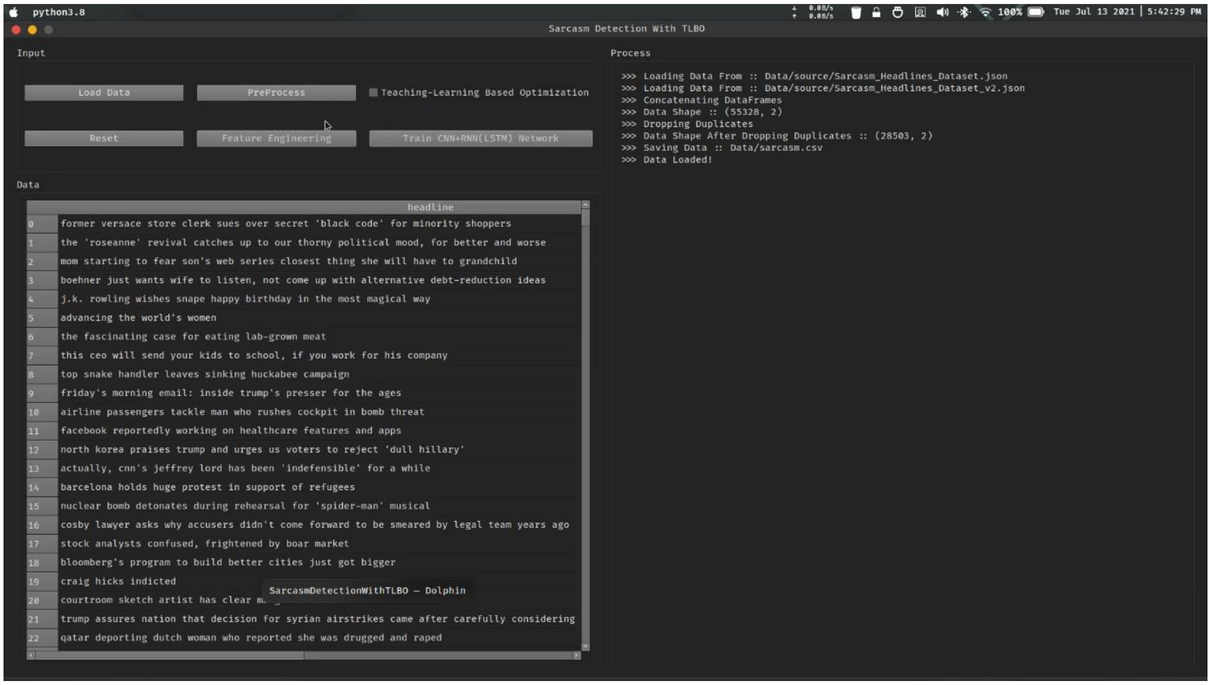
[24] Rao RV, Savsani VJ, Vakharia DP. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Des 2011;43(3):303–15.

[25] Gill, H.S., Khehra, B.S., Singh, A. and Kaur, L., 2019. Teaching-learning-based optimization algorithm to minimize cross entropy for Selecting multilevel threshold values. Egyptian Informatics Journal, 20(1), pp.11-25.

[26] https://www.kaggle.com/rmisra/news-headlines-dataset-for-sarcasm-detection

[27] Ren, L., Xu, B., Lin, H., Liu, X. and Yang, L., 2020. Sarcasm detection with sentiment semantics enhanced multi-level memory network. Neurocomputing, 401, pp.320-326.

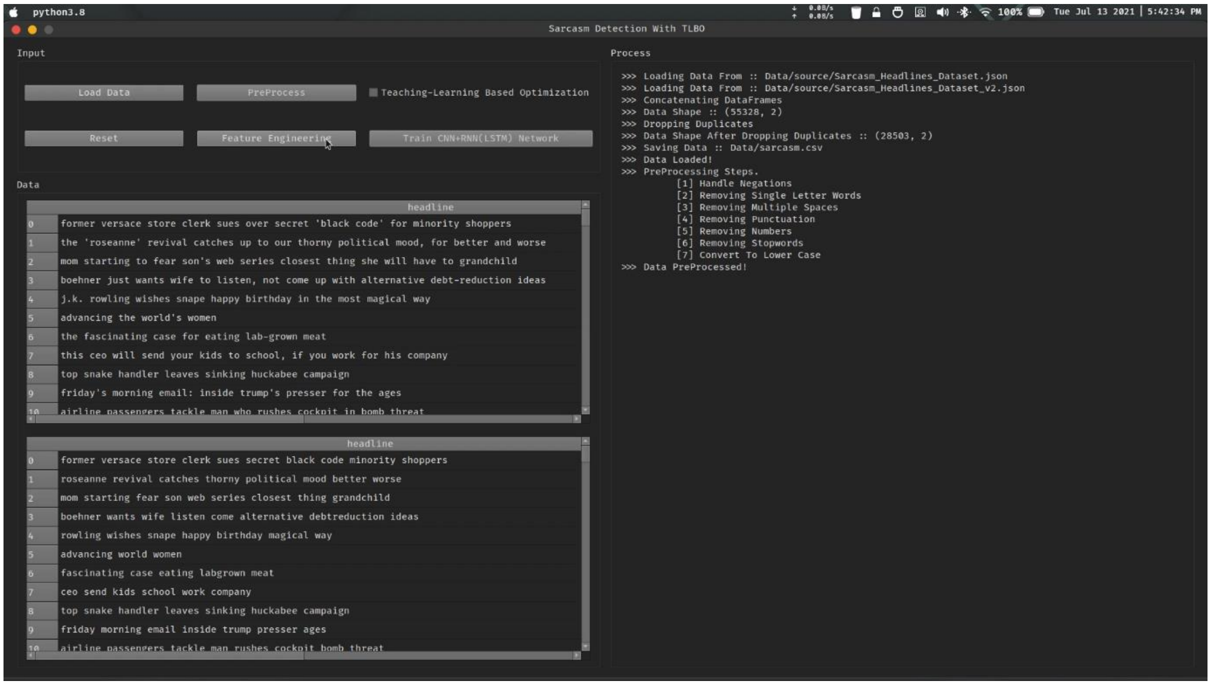
APPENDIX


python3.8
0.0B/s
0.0B/s
100%
Tue Jul 13 2021 | 5:42:29 PM
Sarcasm Detection With TLBO
Input
Load Data
PreProcess
Teaching-Learning Based Optimization
Reset
Feature Engineering
Train CNN+RNN(LSTM) Network
Data
Process
SarcasmDetectionWithTLBO — Dolphin

	headline
0	former versace store clerk sues over secret 'black code' for minority shoppers
1	the 'roseanne' revival catches up to our thorny political mood, for better and worse
2	mom starting to fear son's web series closest thing she will have to grandchild
3	boehner just wants wife to listen, not come up with alternative debt-reduction ideas
4	j.k. rowling wishes snape happy birthday in the most magical way
5	advancing the world's women
6	the fascinating case for eating lab-grown meat
7	this ceo will send your kids to school, if you work for his company
8	top snake handler leaves sinking huckabee campaign
9	friday's morning email: inside trump's presser for the ages
10	airline passengers tackle man who rushes cockpit in bomb threat
11	facebook reportedly working on healthcare features and apps
12	north korea praises trump and urges us voters to reject 'dull hillary'
13	actually, cnn's jeffrey lord has been 'indefensible' for a while
14	barcelona holds huge protest in support of refugees
15	nuclear bomb detonates during rehearsal for 'spider-man' musical
16	cosby lawyer asks why accusers didn't come forward to be smeared by legal team years ago
17	stock analysts confused, frightened by boar market
18	bloomberg's program to build better cities just got bigger
19	craig hicks indicted
20	courtroom sketch artist has clear m
21	trump assures nation that decision for syrian airstrikes came after carefully considering
22	qatar deporting dutch woman who reported she was drugged and raped

```
>>> Loading Data From :: Data/source/Sarcasm_Headlines_Dataset.json
>>> Loading Data From :: Data/source/Sarcasm_Headlines_Dataset_v2.json
>>> Concatenating DataFrames
>>> Data Shape :: (55328, 2)
>>> Dropping Duplicates
>>> Data Shape After Dropping Duplicates :: (28503, 2)
>>> Saving Data :: Data/sarcasm.csv
>>> Data Loaded!
```

Fig. 13. Loading Dataset Module

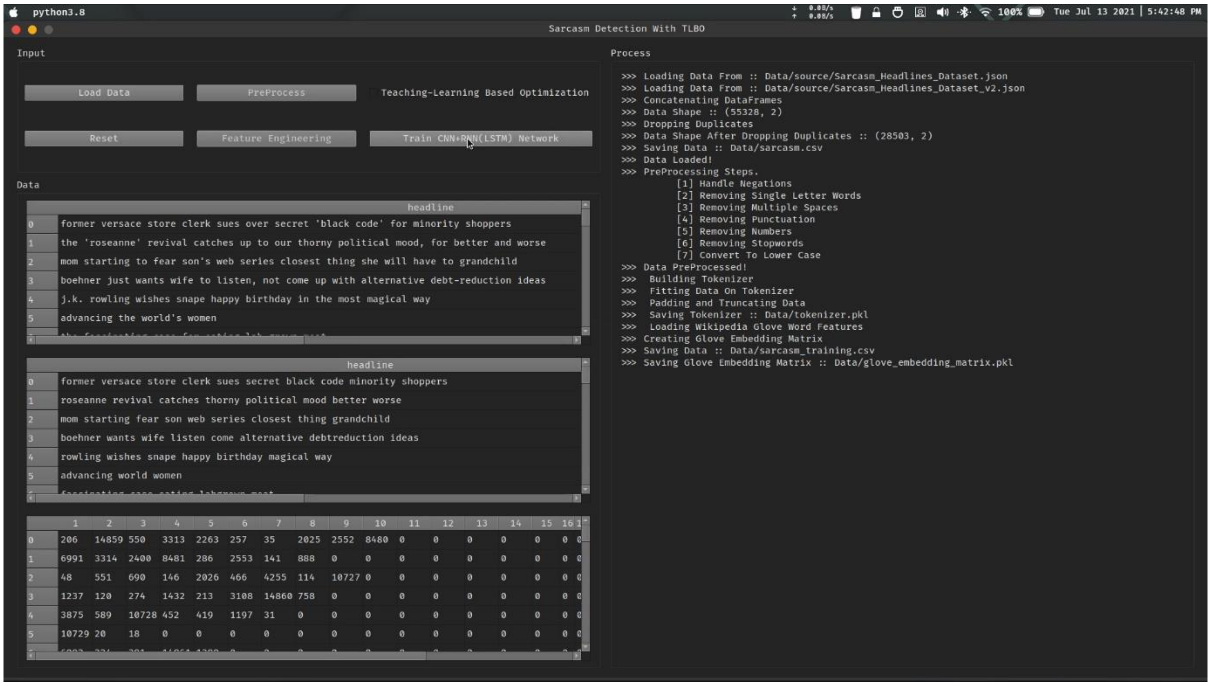

python3.8
0.0B/s
0.0B/s
100%
Tue Jul 13 2021 | 5:42:34 PM
Sarcasm Detection With TLBO
Input
Load Data
PreProcess
□ Teaching-Learning Based Optimization
Reset
Feature Engineering
Train CNN+RNN(LSTM) Network
Data
Process

	headline
0	former versace store clerk sues over secret 'black code' for minority shoppers
1	the 'roseanne' revival catches up to our thorny political mood, for better and worse
2	mom starting to fear son's web series closest thing she will have to grandchild
3	boehner just wants wife to listen, not come up with alternative debt-reduction ideas
4	j.k. rowling wishes snape happy birthday in the most magical way
5	advancing the world's women
6	the fascinating case for eating lab-grown meat
7	this ceo will send your kids to school, if you work for his company
8	top snake handler leaves sinking huckabee campaign
9	friday's morning email: inside trump's presser for the ages
10	airline passengers tackle man who rushes cockpit in bomb threat

	headline
0	former versace store clerk sues secret black code minority shoppers
1	roseanne revival catches thorny political mood better worse
2	mom starting fear son web series closest thing grandchild
3	boehner wants wife listen come alternative debtreduction ideas
4	rowling wishes snape happy birthday magical way
5	advancing world women
6	fascinating case eating labgrown meat
7	ceo send kids school work company
8	top snake handler leaves sinking huckabee campaign
9	friday morning email inside trump presser ages
10	airline passengers tackle man rushes cockpit bomb threat

```
>>> Loading Data From :: Data/source/Sarcasm_Headlines_Dataset.json
>>> Loading Data From :: Data/source/Sarcasm_Headlines_Dataset_v2.json
>>> Concatenating DataFrames
>>> Data Shape :: (55328, 2)
>>> Dropping Duplicates
>>> Data Shape After Dropping Duplicates :: (28503, 2)
>>> Saving Data :: Data/sarcasm.csv
>>> Data Loaded!
>>> PreProcessing Steps.
        [1] Handle Negations
        [2] Removing Single Letter Words
        [3] Removing Multiple Spaces
        [4] Removing Punctuation
        [5] Removing Numbers
        [6] Removing Stopwords
        [7] Convert To Lower Case
>>> Data PreProcessed!
```

Fig. 14. Preprocessing Module


python3.8
0.0B/s
0.0B/s
100%
Tue Jul 13 2021 | 5:42:48 PM
Sarcasm Detection With TLBO
Input
Load Data
PreProcess
Teaching-Learning Based Optimization
Reset
Feature Engineering
Train CNN+RNN(LSTM) Network
Data
Process

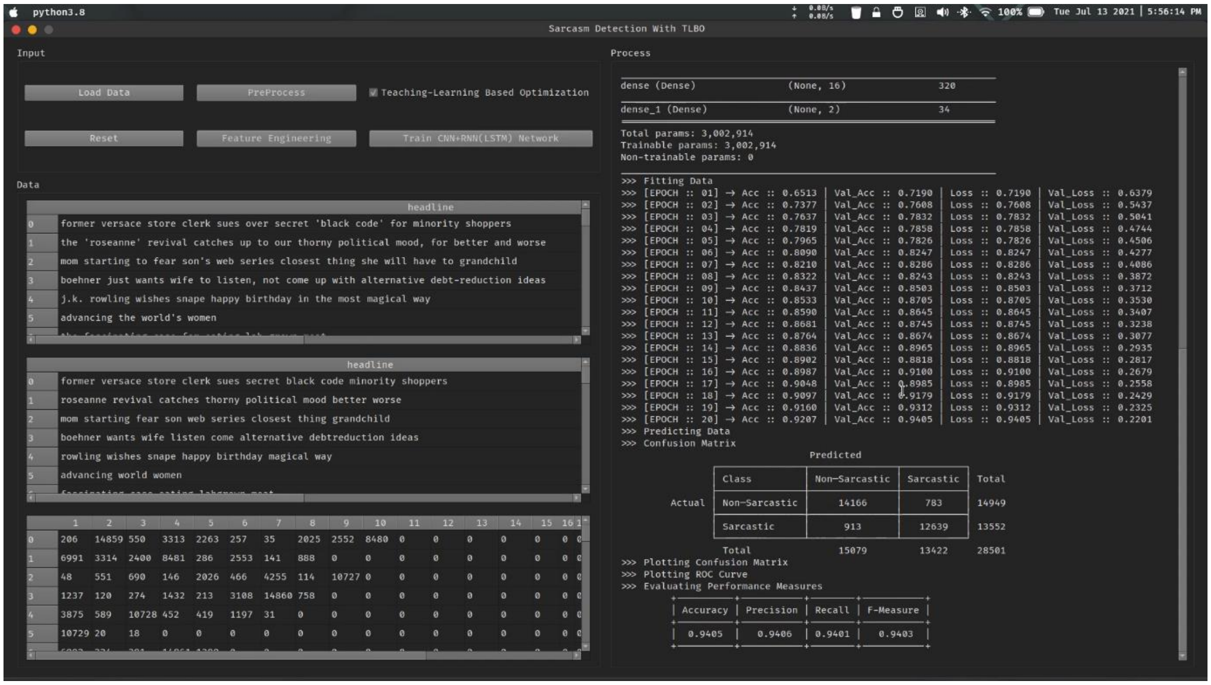
	headline
0	former versace store clerk sues over secret 'black code' for minority shoppers
1	the 'roseanne' revival catches up to our thorny political mood, for better and worse
2	mom starting to fear son's web series closest thing she will have to grandchild
3	boehner just wants wife to listen, not come up with alternative debt-reduction ideas
4	j.k. rowling wishes snape happy birthday in the most magical way
5	advancing the world's women

	headline
0	former versace store clerk sues secret black code minority shoppers
1	roseanne revival catches thorny political mood better worse
2	mom starting fear son web series closest thing grandchild
3	boehner wants wife listen come alternative debtreduction ideas
4	rowling wishes snape happy birthday magical way
5	advancing world women

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	206	14859	550	3313	2263	257	35	2025	2552	8480	0	0	0	0	0	0
1	6991	3314	2400	8481	286	2553	141	888	0	0	0	0	0	0	0	0
2	48	551	690	146	2026	466	4255	114	10727	0	0	0	0	0	0	0
3	1237	120	274	1432	213	3108	14860	758	0	0	0	0	0	0	0	0
4	3875	589	10728	452	419	1197	31	0	0	0	0	0	0	0	0	0
5	10729	20	18	0	0	0	0	0	0	0	0	0	0	0	0	0

```
>>> Loading Data From :: Data/source/Sarcasm_Headlines_Dataset.json
>>> Loading Data From :: Data/source/Sarcasm_Headlines_Dataset_v2.json
>>> Concatenating DataFrames
>>> Data Shape :: (55328, 2)
>>> Dropping Duplicates
>>> Data Shape After Dropping Duplicates :: (28503, 2)
>>> Saving Data :: Data/sarcasm.csv
>>> Data Loaded!
>>> PreProcessing Steps.
        [1] Handle Negations
        [2] Removing Single Letter Words
        [3] Removing Multiple Spaces
        [4] Removing Punctuation
        [5] Removing Numbers
        [6] Removing Stopwords
        [7] Convert To Lower Case
>>> Data PreProcessed!
>>>  Building Tokenizer
>>>  Fitting Data On Tokenizer
>>>  Padding and Truncating Data
>>>  Saving Tokenizer :: Data/tokenizer.pkl
>>>  Loading Wikipedia Glove Word Features
>>> Creating Glove Embedding Matrix
>>> Saving Data :: Data/sarcasm_training.csv
>>> Saving Glove Embedding Matrix :: Data/glove_embedding_matrix.pkl
```

Fig. 15. Feature Extraction Module


python3.8
0.0B/s
0.0B/s
100%
Tue Jul 13 2021 | 5:56:14 PM
Sarcasm Detection With TLBO
Input
Load Data
PreProcess
☑ Teaching-Learning Based Optimization
Reset
Feature Engineering
Train CNN+RNN(LSTM) Network
Data
Process

	headline
0	former versace store clerk sues over secret 'black code' for minority shoppers
1	the 'roseanne' revival catches up to our thorny political mood, for better and worse
2	mom starting to fear son's web series closest thing she will have to grandchild
3	boehner just wants wife to listen, not come up with alternative debt-reduction ideas
4	j.k. rowling wishes snape happy birthday in the most magical way
5	advancing the world's women

	headline
0	former versace store clerk sues secret black code minority shoppers
1	roseanne revival catches thorny political mood better worse
2	mom starting fear son web series closest thing grandchild
3	boehner wants wife listen come alternative debtreduction ideas
4	rowling wishes snape happy birthday magical way
5	advancing world women

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	206	14859	550	3313	2263	257	35	2025	2552	8480	0	0	0	0	0	0
1	6991	3314	2400	8481	286	2553	141	888	0	0	0	0	0	0	0	0
2	48	551	690	146	2026	466	4255	114	10727	0	0	0	0	0	0	0
3	1237	120	274	1432	213	3108	14860	758	0	0	0	0	0	0	0	0
4	3875	589	10728	452	419	1197	31	0	0	0	0	0	0	0	0	0
5	10729	20	18	0	0	0	0	0	0	0	0	0	0	0	0	0

```
dense (Dense)                (None, 16)                320
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 34
=================================================================
Total params: 3,002,914
Trainable params: 3,002,914
Non-trainable params: 0
_________________________________________________________________
>>> Fitting Data
>>> [EPOCH :: 01] → Acc :: 0.6513 | Val_Acc :: 0.7190 | Loss :: 0.7190 | Val_Loss :: 0.6379
>>> [EPOCH :: 02] → Acc :: 0.7377 | Val_Acc :: 0.7608 | Loss :: 0.7608 | Val_Loss :: 0.5437
>>> [EPOCH :: 03] → Acc :: 0.7637 | Val_Acc :: 0.7832 | Loss :: 0.7832 | Val_Loss :: 0.5041
>>> [EPOCH :: 04] → Acc :: 0.7819 | Val_Acc :: 0.7858 | Loss :: 0.7858 | Val_Loss :: 0.4744
>>> [EPOCH :: 05] → Acc :: 0.7965 | Val_Acc :: 0.7826 | Loss :: 0.7826 | Val_Loss :: 0.4506
>>> [EPOCH :: 06] → Acc :: 0.8090 | Val_Acc :: 0.8247 | Loss :: 0.8247 | Val_Loss :: 0.4277
>>> [EPOCH :: 07] → Acc :: 0.8210 | Val_Acc :: 0.8286 | Loss :: 0.8286 | Val_Loss :: 0.4086
>>> [EPOCH :: 08] → Acc :: 0.8322 | Val_Acc :: 0.8243 | Loss :: 0.8243 | Val_Loss :: 0.3872
>>> [EPOCH :: 09] → Acc :: 0.8437 | Val_Acc :: 0.8503 | Loss :: 0.8503 | Val_Loss :: 0.3712
>>> [EPOCH :: 10] → Acc :: 0.8533 | Val_Acc :: 0.8705 | Loss :: 0.8705 | Val_Loss :: 0.3530
>>> [EPOCH :: 11] → Acc :: 0.8590 | Val_Acc :: 0.8645 | Loss :: 0.8645 | Val_Loss :: 0.3407
>>> [EPOCH :: 12] → Acc :: 0.8681 | Val_Acc :: 0.8745 | Loss :: 0.8745 | Val_Loss :: 0.3238
>>> [EPOCH :: 13] → Acc :: 0.8764 | Val_Acc :: 0.8674 | Loss :: 0.8674 | Val_Loss :: 0.3077
>>> [EPOCH :: 14] → Acc :: 0.8836 | Val_Acc :: 0.8965 | Loss :: 0.8965 | Val_Loss :: 0.2935
>>> [EPOCH :: 15] → Acc :: 0.8902 | Val_Acc :: 0.8818 | Loss :: 0.8818 | Val_Loss :: 0.2817
>>> [EPOCH :: 16] → Acc :: 0.8987 | Val_Acc :: 0.9100 | Loss :: 0.9100 | Val_Loss :: 0.2679
>>> [EPOCH :: 17] → Acc :: 0.9048 | Val_Acc :: 0.8985 | Loss :: 0.8985 | Val_Loss :: 0.2558
>>> [EPOCH :: 18] → Acc :: 0.9097 | Val_Acc :: 0.9179 | Loss :: 0.9179 | Val_Loss :: 0.2429
>>> [EPOCH :: 19] → Acc :: 0.9160 | Val_Acc :: 0.9312 | Loss :: 0.9312 | Val_Loss :: 0.2325
>>> [EPOCH :: 20] → Acc :: 0.9207 | Val_Acc :: 0.9405 | Loss :: 0.9405 | Val_Loss :: 0.2201
>>> Predicting Data
>>> Confusion Matrix
                              Predicted
                 ┌──────────────┬──────────────┬────────────┐
                 │ Class        │ Non-Sarcastic│ Sarcastic  │ Total
                 ├──────────────┼──────────────┼────────────┤
         Actual  │ Non-Sarcastic│    14166     │    783     │ 14949
                 ├──────────────┼──────────────┼────────────┤
                 │ Sarcastic    │     913      │   12639    │ 13552
                 └──────────────┴──────────────┴────────────┘
                   Total             15079        13422       28501
>>> Plotting Confusion Matrix
>>> Plotting ROC Curve
>>> Evaluating Performance Measures
        +----------+-----------+--------+-----------+
        | Accuracy | Precision | Recall | F-Measure |
        +----------+-----------+--------+-----------+
        |  0.9405  |   0.9406  | 0.9401 |   0.9403  |
        +----------+-----------+--------+-----------+
```

Fig. 16. Classification Module