SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V10I5P101

Volume 10 Issue 5, 1-13, May 2023

Original Article

Robust Sarcasm Detection using Artificial Rabbits Optimizer with Multilayer Convolutional Encoder-Decoder Neural Network on Social Media

A. Palaniammal[1], P. Anandababu[2]

[1, 2]*Department of Computer and Information Science, Annamalai University.*

[1]*Corresponding Author : revathy20k@gmail.com*

Received: 01 March 2023 Revised: 08 April 2023 Accepted: 02 May 2023 Published: 31 May 2023

Abstract *- Nowadays, posting sarcastic comments on media platforms developed a general trend. People to pester or taunt others frequently utilize sarcasm. It is regularly stated that with tonal stress, inflexion from the speech or in the procedure of hyperbolic, lexical, and pragmatic aspects occur from the textual data. Sarcasm Detection (SD) utilizing Deep Learning (DL) on media platforms is an active study field in Natural Language Processing (NLP). Sarcasm is a figurative language method frequently exploited on social networks like Reddit, Twitter, and Facebook. Detecting sarcasm is essential to various applications like Sentiment Analysis (SA), opinion mining, and social network monitoring. DL techniques are demonstrated that effectual at sarcasm detection on media platforms. This study presents a robust sarcasm detection using Artificial Rabbits Optimizer with Multilayer Convolutional Encoder-Decoder Neural Network (ARO-MCEDNN) technique on social media—the presented ARO-MCEDNN technique concentrations on detecting sarcasm in social networking sites. Primarily, the ARO-MCEDNN technique follows a series of pre-processing data levels for transforming the input data into a compatible format. Followed by, Glove approach is applied for word embedding purposes. Moreover, the MCEDNN model is applied as a classification model to identify and categorize distinct kinds of sarcasm. Furthermore, the ARO algorithm is chosen as a hyperparameter optimizer of the MCEDNN model, enhancing the sarcasm detection performance. To highlight the advanced performance of the ARO-MCEDNN system, a sequence of simulations was performed.*

Keywords *- Social media, Natural language processing, Deep learning, Glove approach, Artificial rabbit optimizer.*

1. Introduction

Sarcasm is widely adopted in social media platforms and microblogging websites in which individuals reprimand or mock, making it hard even for humans to differentiate whether what is told is what is intended [1]. The metaphoric way of sarcasm makes it a frequently cited threat for sentiment investigation. This sarcasm has an indirect pessimist sentiment but an optimistic exterior sentiment [2]. The key challenge and the merits of SD to SA have led to attention to Automatic SD (ASD) as a research issue [3]. ASD denotes estimative methods that foresee whether a specified text is in a sarcastic representation. In the text classification, SD is a critical tool with several suggestions for many fields encompassing health [4], safety, and sales. With the assistance of SD procedures, institutions can examine their customers' emotional state toward their products. This offers significant help for those companies/institutions to enrich their product worth [5]. In SA, the classification of sarcasm is a crucial sub-work, particularly in categorizing tweets, for assigning implicit data within the note that a person conveys or shares among others [6]. Additionally, the system of the tweet may also be implemented in anticipating sarcasm, for example, converting the conflicts of a pessimistic/optimistic remark to its contradictory form. In Twitter, various problems make SA a challenging process.

SD signifies the implementation of DL/ML, Natural Language Processing (NLP), and statistical procedures for distinguishing satire or cynicism positioning for various impurities of a communication, text message, or document layers [7-9]. This is also often dignified as the binary classification issue. The previous research in predicting an ironic statement primarily intended on regulation-based and arithmetic procedures (i) the existence of sentiment alterations, interjections, punctuations, etc., and (ii) pragmatic and lexical structures [10]. A Deep Neural Network provides a process for fundamental crucial learning methods, alternatively implementing hand-engineered methods [11, 12]. The DL procedure has performed a progressive routine in diverse NLP structures like text message narration, responding to queries, and machine translation [13]. This DL procedure has been researched in SD and appears to achieve great motivation.

This study presents a robust sarcasm detection using Artificial Rabbits Optimizer with Multilayer Convolutional Encoder-Decoder Neural Network (ARO-MCEDNN) technique on social media. Primarily, the ARO-MCEDNN technique follows a series of pre-processing data levels to alter the input information into a compatible format. Succeeded by the Glove approach is applied for word embedding purposes. Moreover, the MCEDNN model is applied as a classification model to identify and categorize distinct kinds of sarcasm. Furthermore, the ARO algorithm is chosen as a hyperparameter optimizer of the MCEDNN model, enhancing the sarcasm detection performance. To highlight the improved outcomes of the ARO-MCEDNN system, a sequence of simulations is executed.

2. Related Works

Elkamchouchi et al. [14] present a state-of-the-art (HCOA-SACDC) approach called Hosted Cuckoo Optimizer Algorithm with SAE-Enabled SD and Classification method. This HCOA-SACDC technique mainly focused on the classification and SD in the OSN atmosphere. Likewise, for practical feature extraction, the TF-IDF technique was utilized. Besides, for categorizing and recognizing sarcasm, the SAE model was utilized. Pandey and Singh [15] projected a hybrid BERT-LSTM technique.

A pretrained BERT method was utilized for creating embedding for the code-mixed database. Sharma et al. [16] devised a new hybrid method using an AE called an embedding-based approach. The structure devises using sentence embedding from LSTM-AE, universal sentence encoder and bidirectional encoder representation transformers. The text over images was considered for managing multimedia content like videos and images. The last structure was modelled after an ablation study of different hybrid fusions of methods.

Ren et al. [17] offer multilayer memory networking by employing sentiment semantics to capture sarcastic terminologies' attributes. The author utilised the first-level memory network in this technique for capturing sentiment semantics and the contrast between the situation and sentiment semantics in all sentences. The second-level memory networking is employed. Zhang et al. [18] devised a novel SLSD (stance-level sarcasm detection) task. In contrast, the aim is to expose the latent stances of the author and dependent upon them to find the sarcastic polarity from the textual data. The authors present an innovative stance-centred graph attention network (SCGAT) and an integral structure, which entails BERT.

Zhao et al. [19] offer a Coupled-Attention Network (CAN), which can successfully integrate text information and images as a unified structure, therefore realizing the fusion of various resources.

3. The Proposed Model

In this research, we have presented a new ARO-MCEDNN model for SD in social media. The presented ARO-MCEDNN system emphasizes sarcasm detection and classification in social networking sites. It follows a four-stage process: data pre-processing, Glove based word embedding, MCEDNN-based sarcasm detection, and ARO-based hyperparameter tuning. Fig. 1 expresses the workflow of the ARO-MCEDNN model.

3.1. Data Pre-processing

A primary step, the ARO-MCEDNN technique follows a series of pre-processing data levels to convert the input dataset into a compatible form. Text pre-processing cleans the actual textual data [20]. A textual pre-processed model was crucial for application on NLP tasks. Each text module reached and then pre-processed serves as the vital input elements provided as textual information application. Pre-processed contains many approaches for translating the unique text as a well-determined method: lexical analysis (ignoring case sensitivity, removing punctuations, symbols or special characters, and word tokenization), lemmatization and removal of stop words.

3.2. Glove Word Embedding

In this work, the Glove approach is applied for word embedding purposes. The gloVe is a Stanford-established un-supervised learning approach which generates word vector depictions [21]. GloVe is a word-embedded system which utilizes co-occurrence matrices for computing word resemblances from the corpus and trains a regression approach for learning word vectors which will capture the connections amongst wordings dependent upon frequently they can appear together in the corpus—gloVe mechanism on two essential elements like nearest neighbours and linear sub-structures.

The cosine similarity between two words is utilized for measuring the neighbourhood distance. The cosine similarity defines that the exact two wordings are semantically or linguistically. The resemblance metrics generated in this neighbourhood estimation comprise a single scalar which gauges the connection among two wordings. While two sentences generally have further complex connections than a single number is defined, this simplicity can require that revision. The model needs to connect one or more numbers with the word pair to statistically capture the data desired to distinguish between two exact words.

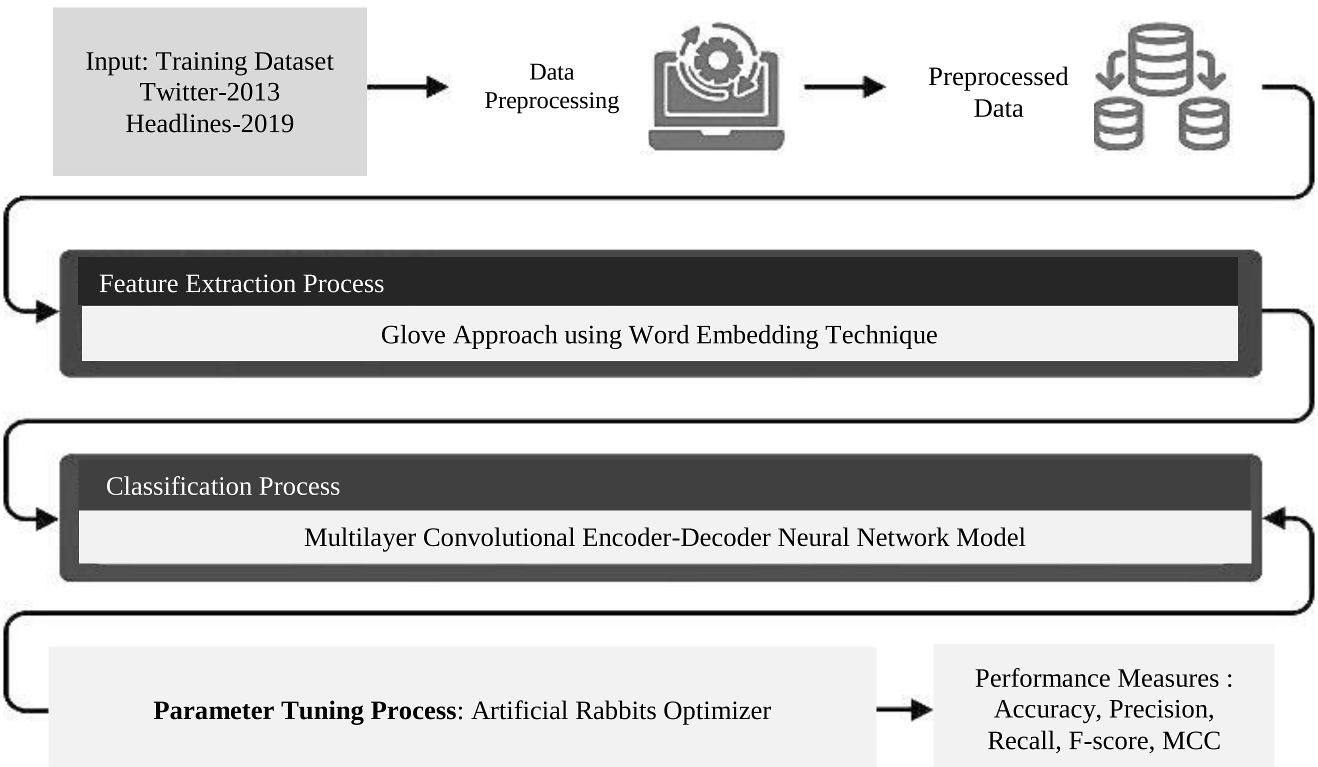

Fig. 1 Workflow of ARO-MCEDNN model

3.3. Sarcasm Detection by Implementing MCEDNN Model

At this stage, the MCEDNN model is applied as a classification model to identify and categorize different types of sarcasm. The MCEDNN approach depended on an encoding-decoding structure with several layers of convolutional and attention mechanisms [22]. Fig. 2 represents the structure of CEDNN. These methods are given training on wordings with rare wording segmentation as subwords. Let us input source sentence S provide as an order of m source tokens, $s_1, \dots, s_m$ and $s_i \in V_s$, whereas V_s implies the source vocabulary. The source token can be embedded from continuous space as $s_1, \dots, s_m$. The embedded $s_i \in \mathbb{R}^d$ is provided as $s_i = w(s_i) + p(i)$, whereas $w(S)$ refers to the word embedded, and $p(i)$ signifies the position embedded equivalent to the position i of token s_i from the source sentence. Both embedded can be acquired in embedding matrices, which are given training and other networking parameters. The encoding and decoding are composed of L layers each. The source token embeddings, $s_1, \dots, s_m$, are linearly mapped for obtaining input vectors of the first encoded layer, $h_1^0, \dots, h_m^0$, in which $h_i^0 \in \mathbb{R}^h$ Moreover, h refer to the input and output dimensional of every encoded and decoded layer. By adding the biases $b \in \mathbb{R}^h$and multiplying a vector with weighted $W \in \mathbb{R}^{h \times d}$ the linear mapping was completed:

$$h_i^0 = W s_i + b \tag{1}$$

During the 1st encoded layer, $2h$ convolution filters of size $3 \times h$ mapping all the sequences of 3 consecutive input vectors to feature vector $f_i^1 \in \mathbb{R}^{2h}$. Paddings (represented by $< pad >$) are supplemented at the start and end of the source sentencing to retain a similar count of resultant vectors as the source tokens.

$$f_i^1 = Conv(h_{i-1}^0, h_i^0, h_{i+1}^0) \tag{2}$$

In Eq. (2), Conv(.) is the convolutional function. This remains a non-linearity utilizing gated linear units (GLU):

$$GLU(f_i^1) = f_{i,1:h}^1 \circ \sigma\left(f_{i,h+1:2h}^1\right) \tag{3}$$

In Eq. (3), $GLU(f_i^1) \in \mathbb{R}^h$, $\circ$ and σ define the component-wise multiplication and sigmoid activation functions correspondingly, and $f_{i,u:v}^1$ signifies the elements of f_i^1 in indices u to v. The input vector to the encoded layer is supplemented as the residual connection. The resultant vector of $the\ l^{th}$ the encoded layer is provided as follows:

$$h_i^l = GLU\left(f_i^l\right) + h_i^{l-1}\ i = 1, \dots, m \tag{4}$$

All the resultant vectors of the last encoded layer, $h_i^L \in \mathbb{R}^h$ was linearly mapped to obtain the encoded resultant vector, $e_i \in \mathbb{R}^d$, utilizing weighted $W_e \in \mathbb{R}^{d \times h}$ and biases $b_e \in \mathbb{R}^d$:

$$e_i = W_e h_i^L + b_e \; i = 1, \dots, m \tag{5}$$

Now, assume the generation of target words t_n at the n^{th} time step in decoded, with $n-1$ target wordings priorly created. All the embedded $t_j \in \mathbb{R}^d$ was linearly mapped to $g_j^0 \in \mathbb{R}^h$ and passed as input to the first decoded layer. During all the decoded layers, convolutional functions followed by non-linearity can be executed on the preceding decoded layer resultant vectors. g_j^{l-1}, whereas $j = 1, \dots, n$:

$$y_j^l = GLU\left(Conv\left(g_{j-3}^{l-1}, g_{j-2}^{l-1}, g_{j-1}^{l-1}\right)\right) \tag{6}$$

In Eq. (6), $Conv(.)$ and GLU correspondingly denote the convolutional and non-linearity, and y_j^l develops the decoded layer at $the\, j^{th}$ time step from the l^{th} decoded layer. The complex filtering's number and dimensions are much similar to the individuals from the encoded. All the decoded layer has their attention component. For computing attention at layer l before forecasting the target token at $the\, n^{th}$ time step, the decoded state $y_n^l \in \mathbb{R}^h$ is linearly mapped to the d-dimension vector with weight $W_z \in \mathbb{R}^{d\times h}$ and bias $b_z \in \mathbb{R}^d$, adding the preceding target tokens embedded:

$$z_n^l = W_z y_n^l + b_z + t_{n-1} \tag{7}$$

The attention weighted $\alpha_{n,i}^l$ can be evaluated by the dot product of encoded output vectors $e_1, \dots, e_m$ with z_n^l and normalizing with softmax:

$$\alpha_{n,i}^l = \frac{\exp\left(e_i^T z_n^l\right)}{\sum_{k=1}^m \exp\left(e_k^T z_n^l\right)} \; i = 1, \dots, m \tag{8}$$

The source context vector x_n^l was evaluated by executing the attention weighted to the encoded output vector summary and the source embedded.

$$x_n^l = \sum_{i=1}^m \alpha_{n,i}^l \left(e_i + s_i\right) \tag{9}$$

Then, the context vector x_n^l is mapped linearly to $c_n^l \in \mathbb{R}^h$. The resultant vector of the l-th decoder layer, g_n^l, denotes the sum of c_n^l, y_n^l, and the preceding layer's resultant vector g_n^{l-1}.

$$g_n^l = y_n^l + c_n^l + g_n^{l-1} \tag{10}$$

The last decoder layer output vector g_n^L is mapped linearly to $d_n \in \mathbb{R}^d$. Dropout is employed at the decoding output, embedding, and prior to each encoding and decoding layer. Then, the vector of the decoding output is mapped to the vocabulary size (V) of the target, and softmax can be evaluated for obtaining the probability of the target word.

$$0_n = W_o d_n + b_o W_o \in \mathbb{R}^{|V_t|\times d}, b_o \in \mathbb{R}^{|V_t|} \tag{11}$$

$$p(t_n = w_i | t_1, \dots, t_{n-1}, S) = \frac{\exp(0_{n,i})}{\sum_{k=1}^{|V_t|} \exp\left(0_{n,k}\right)} \tag{12}$$

Where w_i denotes the i^{th} word in the target vocabulary V_t.

3.4. Hyperparameter Tuning by Implementing ARO Algorithm

At last, the ARO model is chosen as a tuning optimizer of the MCEDNN model, enhancing the sarcasm detection performance. The ARO model stimulates the natural behaviour of rabbits during random hiding and detour foraging [23]. Where $rand$ denotes a random integer within [0,1], using Eq. (13), the initial population can be randomly generated.

$$X = rand \times (UB - LB) + LB \tag{13}$$

In all the iterations, the optimization method is performed using the three fundamental strategies: energy shrinking, detour foraging, and random hiding. The detour foraging method concentrates on discovering novel solutions and, through using (14) to (18), is mathematically modelled. If $f(X_i^{t+1})$ is lesser than (X_i^t), the solution position X_i^t is replaced with X_i^{t+1}. $f(X_i)$ characterizes the fitness value corresponding to the solution X_i.

$$\chi_i^{t+1} = X_i^t + R.\left(X_i^t - X_j^t\right) + round\left(0.5.(0.05 + r_1)\right).n_1 \tag{14}$$

$$R = \left(e - e^{\left(\frac{t-1}{T}\right)^2}\right) \times sin(2\pi r_2) \tag{15}$$

$$c = \begin{cases} 1, if\; k == g(l) \\ 0, else \end{cases} k = 1, \cdots\cdots, d \; and \; l = 1, \cdots\cdots. [r_3.d] \tag{16}$$

$$g = randperm(d) \tag{17}$$

$$n_1 N(0,1) \tag{18}$$

$$f_i^t = \begin{cases} f_i^{t+1}, iff_i^{t+1} < f_i^t \\ f_i^t, iff_i^{t+1} \geq f_i^t \end{cases} \tag{19}$$

$$X_i^r = \begin{cases} X_i^{t+1}, iff_i^{t+1} < f_i^t \\ X_i^r, iff_i^{t+1} \geq f_i^t \end{cases} \tag{20}$$

Where X_i^{t+1} denotes the next location of the present solution X_i^t. r_1, r_2 and r_3 indicate random numbers within [0,1]. T denotes the maximal amount of iterations, and d and n denote the problem dimension and the population size, correspondingly.

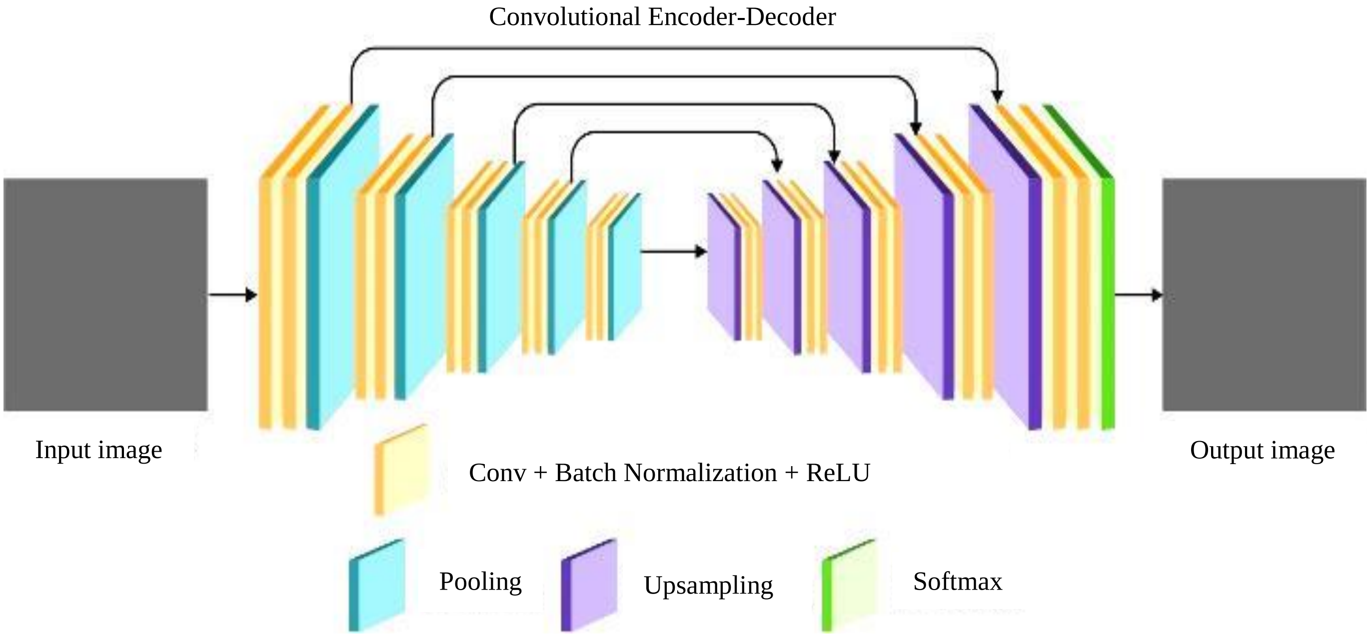

Fig. 2 Structure of CEDNN

Note that the ceiling function, round point to rounding towards the nearest integer, and $randperm$ specifies an integer permutation from 1 to d arbitrarily. The next phase is expressed by Eq. (21) to (23). If (X_i^{t+1}) is lesser than (X_i^t), the solution position X_i^t is replaced with X_i^{t+1}

$$X_i^{t+1} = X_i^t + R \times \left(r_4 \times b_{i,r}^t - X_i^t\right) \tag{21}$$

$$b_{i,r} = \chi_i^t + \frac{T-t+1}{T} . r_4 . g_r . X_i^r \tag{22}$$

$$g_r = \begin{cases} 1, if\ k == [r_5 . d] , l = 1, \cdots \cdots ., d \\ 0, else \end{cases} \tag{23}$$

Where b_{ir}^t signifies a burrow chosen randomly wise from d burrows for hiding. r_4 and r_5 indicate random numbers between zero and one.

Lastly, the energy shrink phase balances the exploration and exploitation features. r denotes the random integer within [0, 1]. When $A > 1$, the exploration stage is carried out. Alternatively, else, the exploitation phase will be applied.

$$A = 4\left(1 - \frac{t}{T}\right) ln\left(\frac{1}{r}\right) \tag{24}$$

When the exploration and exploitation phases are applied, the location of the better solution is upgraded when $f(X_i^{t+1})$ is greater than (X_{best}^t), as follows.

$$f_{best}^t = \begin{cases} f_i^{t+1}, if\ f_i^{t+1} < f_{best}^t \\ f_{besf}^t, if\ f_i^{t+1} \geq f_{best}^t \end{cases} \tag{25}$$

$$X_{besf}^r = \begin{cases} X_i^{t+1}, if\ f_i^{t+1} < f_{best}^t \\ X_{best}^t, if\ f_i^{t+1} \geq f_{best}^t \end{cases} \tag{26}$$

Algorithm 1**:** Pseudocode of ARO Algorithm

Define population size (N_p), problem dimensions (D), upper and lower boundaries, and stop criterion (T).
Set ARO parameter.
Population initialization (X).
Calculate cost function (1).
Allocate a solution that violates the constraint with a higher penalty value.
Describe better and worse solutions.
For r=2 : T
For i=I : N_p
Compute A by utilizing (24).
If A>1
Determine the novel location of every individual (X_i^(t+1)) in X based on exploring ARO Eqs. (14) to (18).
Compute fitness values of novel solutions (f_i^(t+1)).
Assure the constraint and include a cost of penalty if the novel solution interrupts the constraint.
Upgrade individual locations and the better solution location.
Else
Describe the novel location of every individual (X_i^(r+1)) in X based on exploring ARO's Eqs (21) to (23).
Compute fitness values of new solutions (f_i^(r+1)).

Ensure problem constraint and include the penalty cost if the novel solution disrupts the constraint.
Upgrade the individual location and the better solution location.
End if
End for
End for

The selection of fitness function is a critical feature in the ARO model. Exploiting is done for solution encoding to evaluate the aptitude (goodness) of the candidate solution. Later, the value of accuracy is the primary condition used for the FF design.

$$Fitness = \max(P) \tag{27}$$

$$P = \frac{TP}{TP+FP} \tag{28}$$

Whereas TP and FP signify the true and false positive values.

4. Experimental Validation

In this study, the SD outputs of the ARO-MCEDNN approach can be examined on two datasets: Twitter 2013 dataset [24, 25] with 1956 samples and the headlines-2019 dataset [26, 27] with 28619 samples, as denoted in Table 1. The confusion matrix of the ARO-MCEDNN method on the sarcasm detection process under the Twitter 2013 dataset is given in Fig. 3. The results indicate that the ARO-MCEDNN technique correctly identifies sarcastic and non-sarcastic sweets. For instance, with 70% of TRP, the ARO-MCEDNN technique recognizes 212 sarcastic and 1134 non-sarcastic tweets. Additionally, with 30% of TSP, the ARO-MCEDNN approach recognizes 83 sarcastic and 496 non-sarcastic tweets. At last, with 80% of TRP, the ARO-MCEDNN system recognizes 242 tweets with sarcasm and 1305 without sarcasm. Table 2 gives a comprehensive sarcasm recognition outcome of the ARO-MCEDNN technique under the Twitter 2013 dataset. The results imply the ARO-MCEDNN technique's effectual ability to recognise sarcastic and non-sarcastic samples. For instance, with 70% of TRP, the ARO-MCEDNN approach obtains average $accu_y$ of 97.53%, $prec_n$ of 96.35%, $reca_l$ of 97.53%, F_{score} of 96.93%, and MCC of 93.87%. Meanwhile, with 30% of TSS, the ARO-MCEDNN approach gains average $accu_y$ of 96.86%, $prec_n$ of 97.76%, $reca_l$ of 96.86%, F_{score} of 97.30%, and MCC of 94.61%. Eventually, with 80% of TRP, the ARO-MCEDNN method achieves average $accu_y$ of 97.75%, $prec_n$ of 98.21%, $reca_l$ of 97.75%, F_{score} of 97.98%, and MCC of 95.96%. Finally, with 20% of TSP, the ARO-MCEDNN method attains average $accu_y$ of 98.96%, $prec_n$ of 98.96%, $reca_l$ of 98.96%, F_{score} of 98.96%, and MCC of 97.92%.

The TACY and VACY of the ARO-MCEDNN system on the Twitter 2013 dataset are illustrated in Fig. 4. The figure demonstrated that the ARO-MCEDNN algorithm had depicted more significant results with enhanced TACY and VACY values. Note that the ARO-MCEDNN technique has gained maximal TACY outputs. The TLOS and VLOS of the ARO-MCEDNN system on the Twitter 2013 dataset are represented in Fig. 5. The figure illustrates that the ARO-MCEDNN algorithm has demonstrated greater results with lesser TLOS and VLOS values. Note that the ARO-MCEDNN system has gained minimum VLOS outputs.

The confusion matrix of the ARO-MCEDNN system on the sarcasm detection process under the Headline 2019 dataset is represented in Fig. 6. The outputs indicated that the ARO-MCEDNN algorithm correctly identifies sarcastic and non-sarcastic sweets. For instance, with 70% of TRP, the ARO-MCEDNN method recognizes 9381 sarcastic and 10437 non-sarcastic tweets. Additionally, with 30% of TSP, the ARO-MCEDNN approach recognizes 4046 sarcastic and 4446 non-sarcastic tweets. At last, with 80% of TRP, the ARO-MCEDNN method recognizes 10845 sarcastic and 11924 non-sarcastic tweets.

Table 3 gives an overall sarcasm recognition outcome of the ARO-MCEDNN algorithm under the Headline 2019 dataset. The outcomes inferred the ARO-MCEDNN system's effectual ability in recognising sarcastic and non-sarcastic samples. For instance, with 70% of TRP, the ARO-MCEDNN approach gains average $accu_y$ of 98.90%, $prec_n$ of 98.95%, $reca_l$ of 98.90%, F_{score} of 98.92%, and MCC of 97.85%. In the meantime, with 30% of TSS, the ARO-MCEDNN approach gains average $accu_y$ of 98.89%, $prec_n$ of 98.91%, $reca_l$ of 98.89%, F_{score} of 98.90%, and MCC of 97.81%. Afterwards, with 80% of TRP, the ARO-MCEDNN approach gains average $accu_y$ of 99.45%, $prec_n$ of 99.45%, $reca_l$ of 99.45%, F_{score} of 99.45%, and MCC of 98.90%. Finally, with 20% of TSP, the ARO-MCEDNN algorithm gains average $accu_y$ of 99.36%, $prec_n$ of 99.35%, $reca_l$ of 99.36%, F_{score} of 99.35%, and MCC of 98.70%.

The TACY and VACY of the ARO-MCEDNN methodology on the Headline 2019 dataset are illustrated in Fig. 7. The figure represented that the ARO-MCEDNN model has illustrated higher outputs with enhanced TACY value and VACY value. It can be noticeable that the ARO-MCEDNN model has obtained increased TACY outputs. The TLOS value and VLOS value of the ARO-MCEDNN methodology on the Headline 2019 dataset are represented in Fig. 8. The figure pointed out that the ARO-MCEDNN model has illustrated higher results with reduced TLOS and VLOS values. It can be visible that the ARO-MCEDNN model has obtained decreased VLOS outputs.

Finally, the sarcasm detection outputs of the ARO-MCEDNN approach are related to existing methods in Table 4. Fig. 9 represents relative research of the ARO-MCEDNN approach in terms of $accu_y$. The outputs indicate that the ARO-MCEDNN technique reaches effectual results over compared methods. Based on $accu_y$, the ARO-MCEDNN technique gains improved $accu_y$ of 99.45% while the Vanilla-CNN, Fracking sarcasm, ELMo-BiLSTM, ELMo-BiLSTM FULL, and A2Text-Net models accomplish reducing $accu_y$ of 72.63%, 90.15%, 77.06%, 78.49%, and 93.39% respectively.

Fig. 10 shows relative research of the ARO-MCEDNN method in terms of $prec_n$, $reca_l$, and F_{score}. The outcomes stated that the ARO-MCEDNN technique reaches effectual results over compared techniques. Based on $prec_n$, the ARO-MCEDNN method gains improved $prec_n$ of 99.45%, while the Vanilla-CNN, Fracking sarcasm, ELMo-BiLSTM, ELMo-BiLSTM FULL, and A2Text-Net approaches accomplish reducing $prec_n$ of 71.48%, 88.74%, 76.34%, 78.02%, and 92.41% correspondingly. As well, concerning $reca_l$, the ARO-MCEDNN approach obtains an improved $reca_l$ of 99.45% while the Vanilla-CNN, Fracking sarcasm, ELMo-BiLSTM, ELMo-BiLSTM FULL, and A2Text-Net methods accomplish reducing $reca_l$ of 67.76%, 88.63%, 75.77%, 74.22%, and 91.47% correspondingly. Finally, based on F_{score}, the ARO-MCEDNN system acquires improved F_{score} of 99.45%, while the Vanilla-CNN, Fracking sarcasm, ELMo-BiLSTM, ELMo-BiLSTM FULL, and A2Text-Net methods accomplish reducing F_{score} of 68.88%, 88.54%, 76.13%, 75.99%, and 90.53% correspondingly. These outputs reassured the supremacy of the ARO-MCEDNN method over other existing techniques.

Table 1. Dataset details

Class	Data Set	
	Twitter-2013	Headlines-2019
Sarcastic	308	13634
Non Sarcastic	1648	14985
Total No. of Samples	1956	28619

Table 2. Sarcasm recognition outcome of ARO-MCEDNN approach on twitter 2013 data

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}	MCC
Training Phase (70%)					
Sarcastic	96.36	93.39	96.36	94.85	93.87
Non Sarcastic	98.69	99.30	98.69	99.00	93.87
Average	97.53	96.35	97.53	96.93	93.87
Testing Phase (30%)					
Sarcastic	94.32	96.51	94.32	95.40	94.61
Non Sarcastic	99.40	99.00	99.40	99.20	94.61
Average	96.86	97.76	96.86	97.30	94.61
Training Phase (80%)					
Sarcastic	96.03	97.19	96.03	96.61	95.96
Non Sarcastic	99.47	99.24	99.47	99.35	95.96
Average	97.75	98.21	97.75	97.98	95.96
Testing Phase (20%)					
Sarcastic	98.21	98.21	98.21	98.21	97.92
Non Sarcastic	99.70	99.70	99.70	99.70	97.92
Average	98.96	98.96	98.96	98.96	97.92

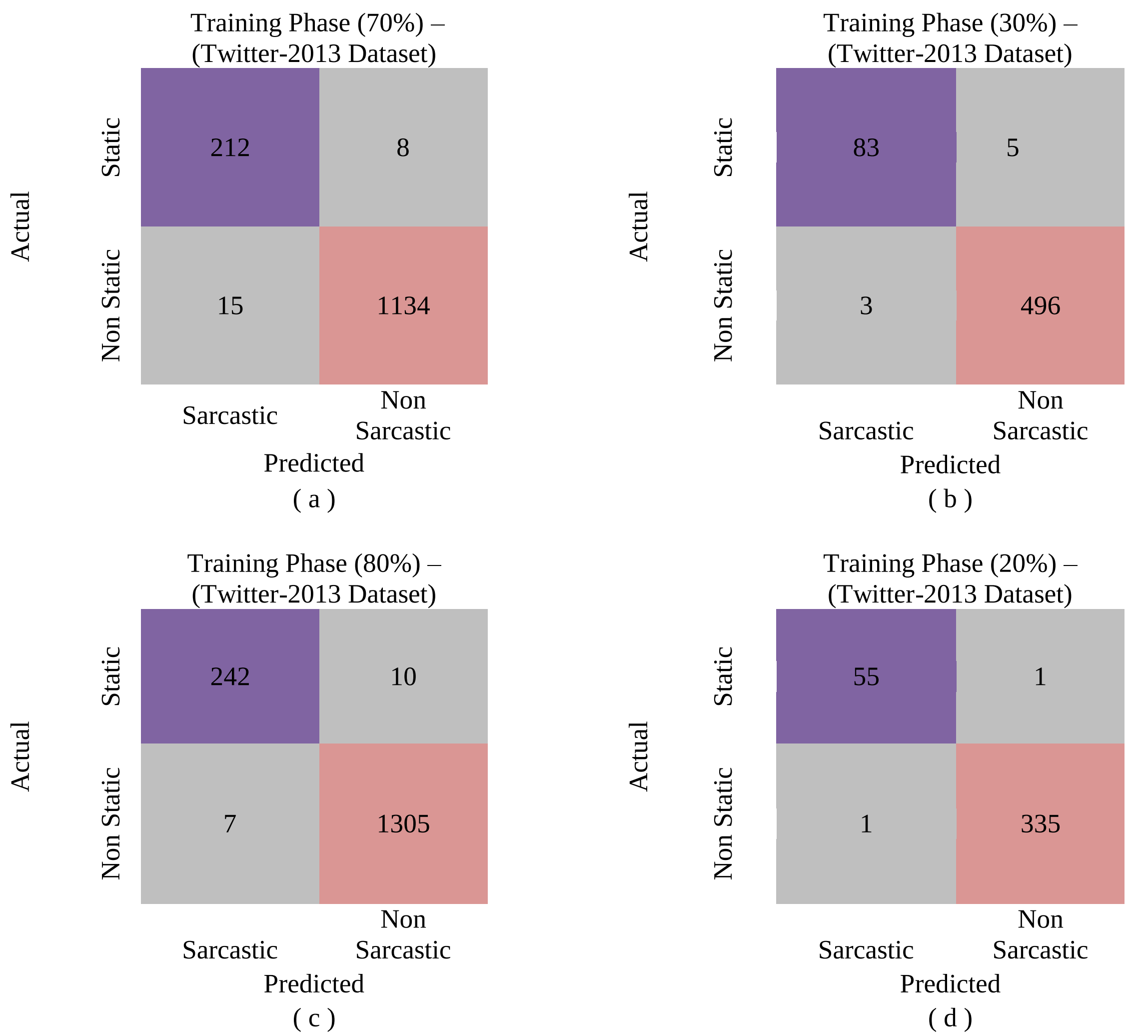

Fig. 3 Confusion matrix of twitter 2013 dataset (a-b) TRP/TSP of 70:30 and (c-d) TRP/TSP of 80:20

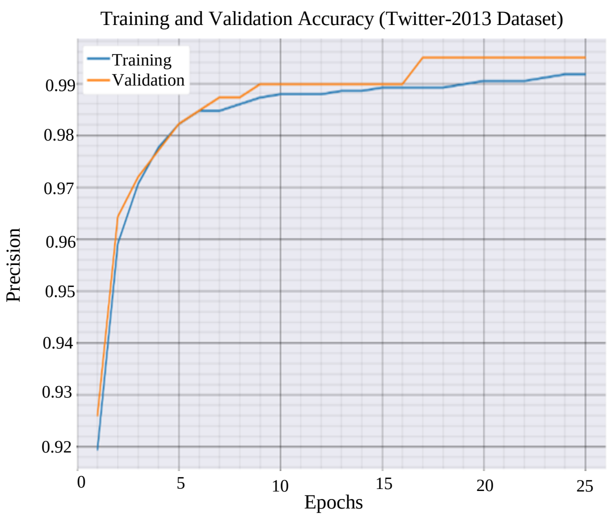

Fig. 4 TACY and VACY outcome of ARO-MCEDNN method on twitter 2013 dataset

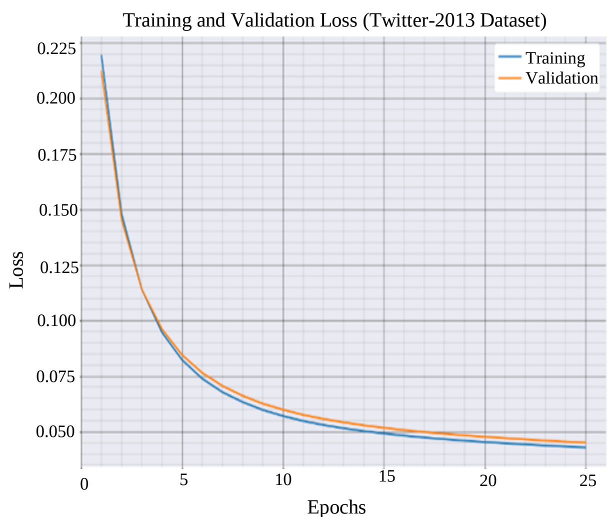

Fig. 5 TLOS and VLOS outcome of ARO-MCEDNN method on twitter 2013 dataset

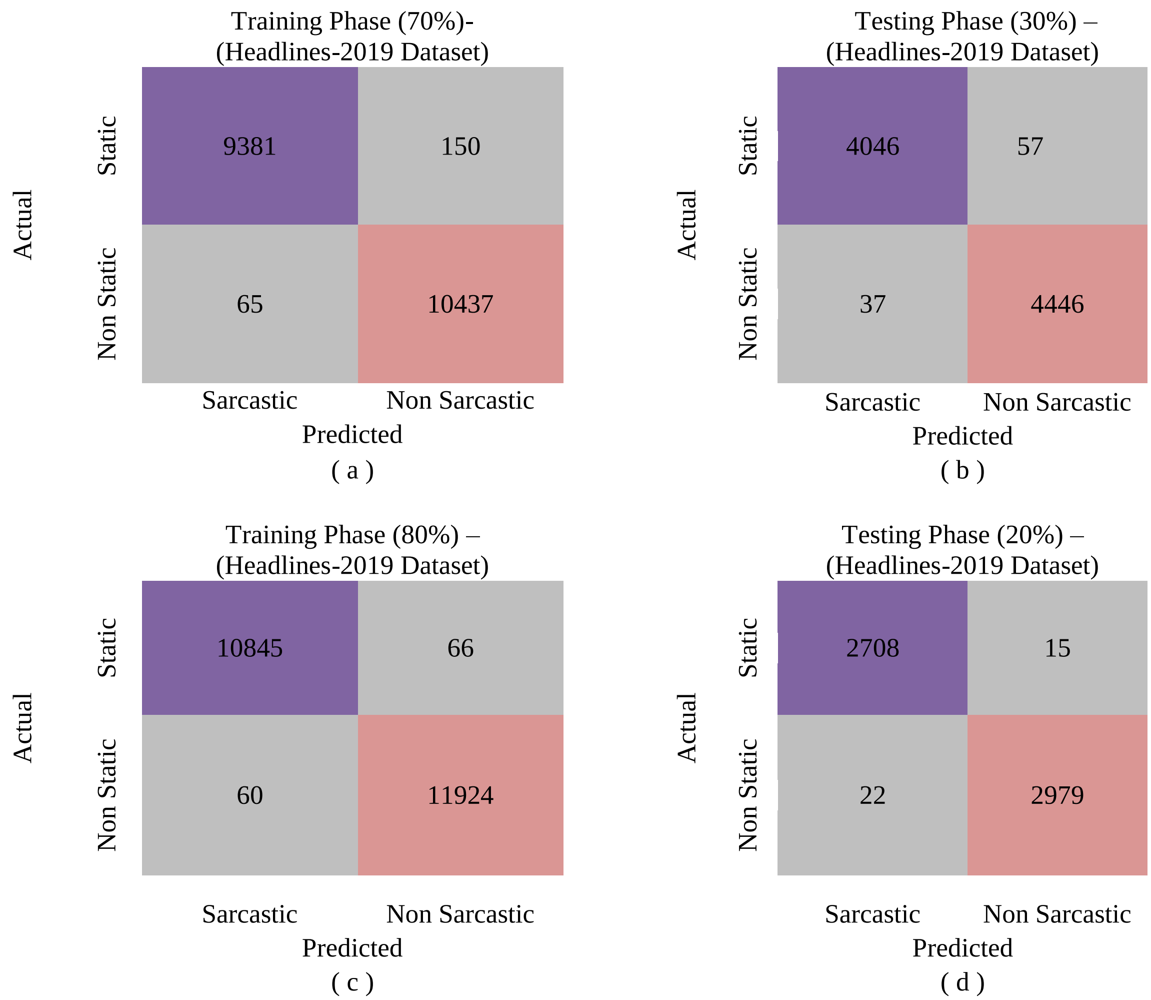

Fig. 6 Confusion matrix of headline 2019 dataset (a-b) TRP/TSP of 70:30 and (c-d) TRP/TSP of 80:20

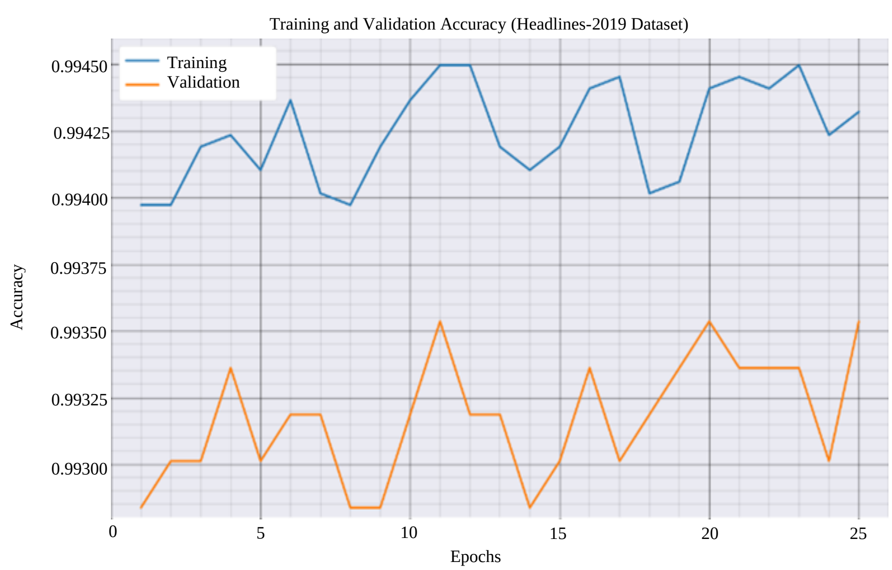

Fig. 7 TACY and VACY outcome of ARO-MCEDNN method on Headline 2019 dataset

Table 3. Sarcasm recognition outcome of ARO-MCEDNN method on headline 2019 dataset

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}	MCC
Training Phase (70%)					
Sarcastic	98.43	99.31	98.43	98.87	97.85
Non Sarcastic	99.38	98.58	99.38	98.98	97.85
Average	98.90	98.95	98.90	98.92	97.85
Testing Phase (30%)					
Sarcastic	98.61	99.09	98.61	98.85	97.81
Non Sarcastic	99.17	98.73	99.17	98.95	97.81
Average	98.89	98.91	98.89	98.90	97.81
Training Phase (80%)					
Sarcastic	99.40	99.45	99.40	99.42	98.90
Non Sarcastic	99.50	99.45	99.50	99.47	98.90
Average	99.45	99.45	99.45	99.45	98.90
Testing Phase (20%)					
Sarcastic	99.45	99.19	99.45	99.32	98.70
Non Sarcastic	99.27	99.50	99.27	99.38	98.70
Average	99.36	99.35	99.36	99.35	98.70

Table 4. The relative outcome of the ARO-MCEDNN method with current systems

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
ARO-MCEDNN	99.45	99.45	99.45	99.45
Vanilla-CNN	72.63	71.48	67.76	68.88
Fracking Sarcasm	90.15	88.74	88.63	88.54
ELMo-BiLSTM	77.06	76.34	75.77	76.13
ELMo-BiLSTM FULL	78.49	78.02	74.22	75.99
A2Text-Net	93.39	92.41	91.47	90.53

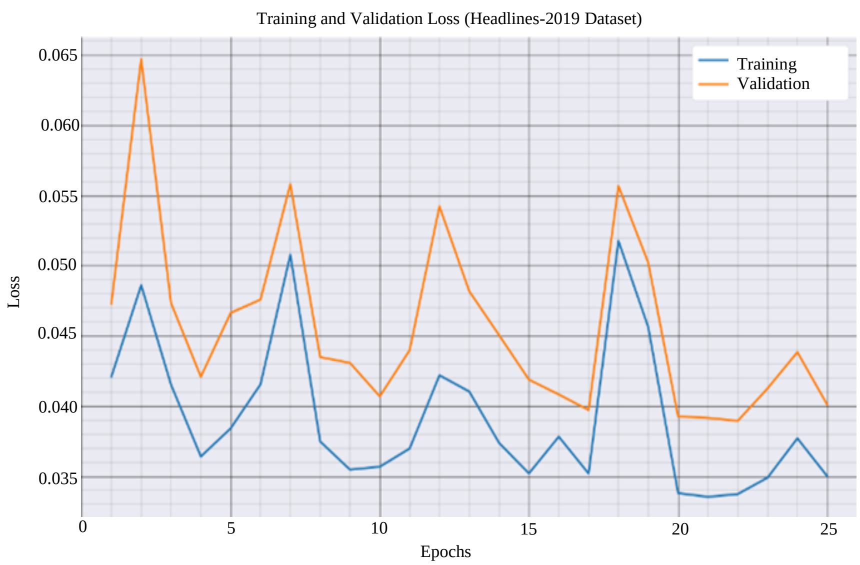

Fig. 8 TLOS and VLOS outcome of ARO-MCEDNN approach on Headline 2019 dataset

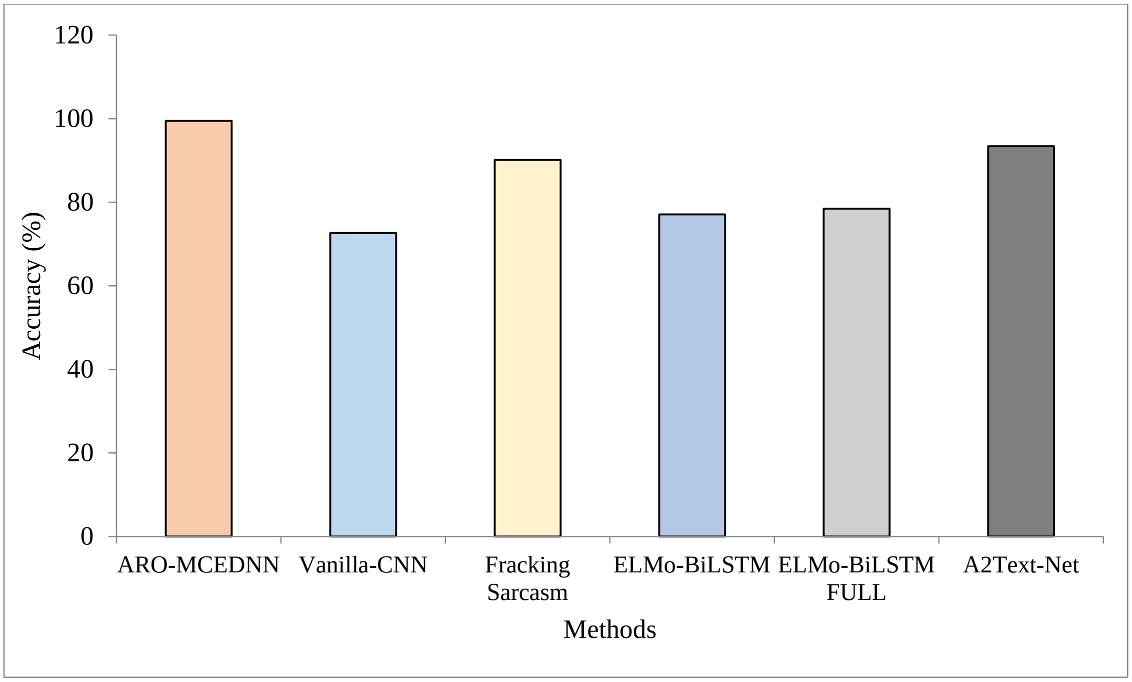

Fig. 9 $Accu_y$ the outcome of the ARO-MCEDNN method with current systems

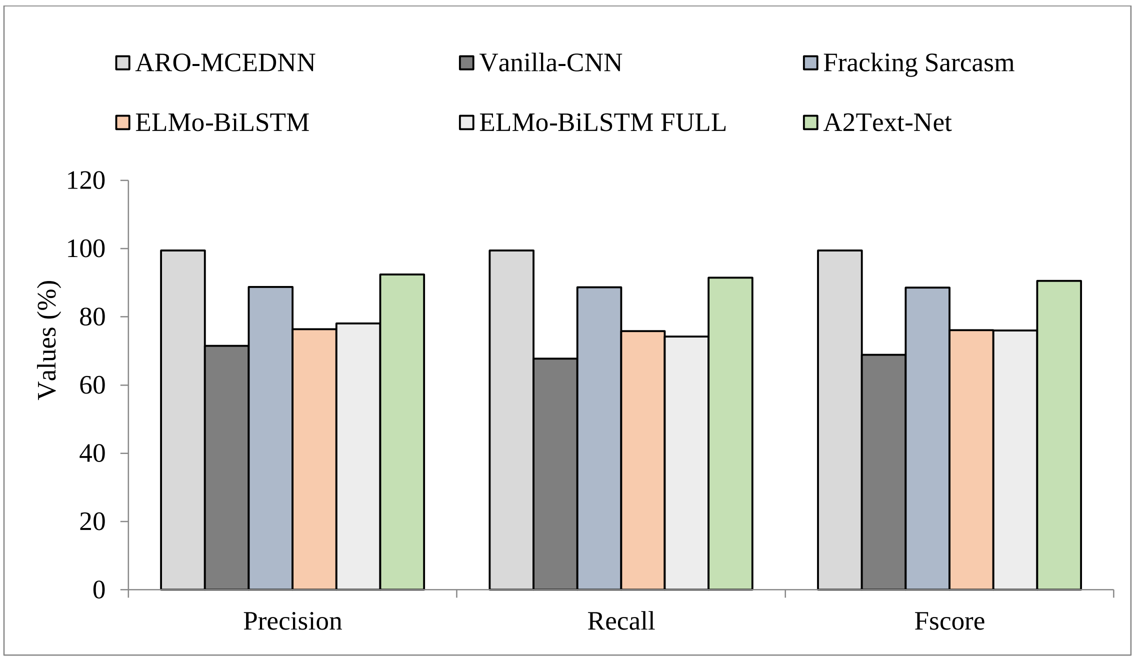

Fig. 10 Relative outcome of ARO-MCEDNN method with current systems

5. Conclusion

In this research, we have presented a new ARO-MCEDNN algorithm for SD in social-media platforms. The proposed ARO-MCEDNN algorithm emphasizes sarcasm detection and classification in social networking sites. It follows a four-stage process: data pre-processing, Glove based word embedding, MCEDNN-based sarcasm detection, and ARO-based hyperparameter tuning. Primarily, the ARO-MCEDNN technique follows a series of pre-processing data levels for converting the input dataset into a compatible format. Succeeded by the Glove approach is applied for word embedding purposes.

Moreover, the MCEDNN model is applied as a classification model to identify and categorize different types of sarcasm. Furthermore, the ARO algorithm is chosen as a hyperparameter optimizer of the MCEDNN model, enhancing the sarcasm detection performance. A series of simulations were performed to accentuate the enhanced performance of the ARO-MCEDNN algorithm. The simulation outputs demonstrate the significant outcomes of the ARO-MCEDNN algorithm over current methods. In future, ensemble DL techniques can be derived to boost the detection performance of the ARO-MCEDNN algorithm.

References

[1] D. Vinoth, and P. Prabhavathy, "An Intelligent Machine Learning-Based Sarcasm Detection and Classification Model on Social Networks," *The Journal of Supercomputing*, vol. 78, no. 8, pp. 10575-10594, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Devamanyu Hazarika et al., "Cascade: Contextual Sarcasm Detection in Online Discussion Forums," *arXiv preprint arXiv:1805.06413*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Yitao Cai, Huiyu Cai, and Xiaojun Wan, "Multi-Modal Sarcasm Detection in Twitter with Hierarchical Fusion Model," *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2506-2515, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Bhumi Shah, and Margil Shah, "A Survey on Machine Learning and Deep Learning Based Approaches for Sarcasm Identification in Social Media," *In Data Science and Intelligent Applications: Proceedings of ICDSIA*, pp. 247-259, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Anbarasu Sivalingam, Karthik Sundararajan, and Anandhakumar Palanisamy, "CRF-MEM: Conditional Random Field Model Based Modified Expectation Maximization Algorithm for Sarcasm Detection in Social Media," *Journal of Internet Technology*, vol. 24, no. 1, pp. 45-54, 2023. [Google Scholar] [Publisher Link]

[6] Poulami Dutta, and Chandan Kumar Bhattacharyya, "Multi-Modal Sarcasm Detection in Social Networks: A Comparative Review," In *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 207-214, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Anuraj Mohan et al., "Sarcasm Detection using Bidirectional Encoder Representations from Transformers and Graph Convolutional Networks," *Procedia Computer Science*, vol. 218, pp. 93-102, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] M. A. Bhalekar, and M. V. Bedekar, "Review on Latest Approaches Used in Natural Language Processing for Generation of Image Captioning," *SSRG International Journal of Computer Science and Engineering*, vol. 4, no. 6, pp. 41-48, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[9] Uthkarsha Sagar, "A Broad Survey of Natural Language Processing," *SSRG International Journal of Computer Science and Engineering*, vol. 6, no. 12, pp. 15-18, 2019. [CrossRef] [Publisher Link]

[10] Md Saifullah Razali et al., "Sarcasm Detection Using Deep Learning with Contextual Features," *IEEE Access*, vol. 9, pp. 68609-68618, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] S. Suma Christal Mary et al., "Selfish Herd Optimization with Improved Deep Learning Based Intrusion Detection for Secure Wireless Sensor Network," *SSRG International Journal of Electronics and Communication Engineering*, vol. 10, no. 4, pp. 1-8, 2023. [Publisher Link]

[12] C. Narmadha et al., "Cloud-based Detection of Malware and Software Privacy Threats in Internet of Things using Deep Learning Approach," *SSRG International Journal of Electronics and Communication Engineering*, vol. 10, no. 4, pp. 21-30, 2023. [Publisher Link]

[13] Yik Yang Tan et al., "Sentiment Analysis and Sarcasm Detection using Deep Multi-Task Learning," *Wireless Personal Communications*, vol. 129, no. 3, pp. 2213-2237, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Dalia H. Elkamchouchi et al., "Hosted Cuckoo Optimization Algorithm with Stacked Autoencoder-Enabled Sarcasm Detection in Online Social Networks," *Applied Sciences*, vol. 12, no. 14, pp. 7119, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Rajnish Pandey, and Jyoti Prakash Singh, "BERT-LSTM Model for Sarcasm Detection in Code-Mixed Social Media Post," *Journal of Intelligent Information Systems*, vol. 60, no. 1, pp. 235-254, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Dilip Kumar Sharma et al., "Sarcasm Detection Over Social Media Platforms using Hybrid Auto-Encoder-Based Model," *Electronics*, vol. 11, no. 18, pp. 2844, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Lu Ren et al., "Sarcasm Detection with Sentiment Semantics Enhanced Multi-Level Memory Network," *Neurocomputing*, vol. 401, pp. 320-326, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Yazhou Zhang et al., "Stance Level Sarcasm Detection with BERT and Stance-Centered Graph Attention Networks," *ACM Transactions on Internet Technology (TOIT)*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Xuan Zhao, Jimmy Huang, and Haitian Yang, "CANs: Coupled-Attention Networks for Sarcasm Detection on Social Media," *In 2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Dakshnamoorthy Vinoth, and Panneer Prabhavathy, "Automated Sarcasm Detection and Classification Using Hyperparameter Tuned Deep Learning Model for Social Networks," *Expert Systems*, vol. 39, no. 10, pp. 13107, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Dilip Kumar Sharma et al., "Sarcasm Detection over Social Media Platforms Using Hybrid Ensemble Model with Fuzzy Logic," *Electronics*, vol. 12, no. 4, pp. 937, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Shamil Chollampatt, and Hwee Tou Ng, "A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction," *In Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[23] Mahendiran Vellingiri et al., "Maximum Hosting Capacity Estimation for Renewables in Power Grids Considering Energy Storage and Transmission Lines Expansion Using Hybrid Sine Cosine Artificial Rabbits Algorithm," *Ain Shams Engineering Journal*, vol. 14, no. 5, pp. 102092, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] [Online]. Available: https://www.kaggle.com/competitions/gse002/data

[25] Ellen Riloff et al., "Sarcasm as Contrast Between a Positive Sentiment and Negative Situation," *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 704–714, 2013. [Google Scholar] [Publisher Link]

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

[27] Rishabh Misra, and Prahal Arora, "Sarcasm Detection using Hybrid Neural Network," *arXiv:1908.07414*, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[28] Ayush Jain et al., "Detection of Sarcasm Through Tone Analysis on Video and Audio Files: A Comparative Study on AI Models Performance," *SSRG International Journal of Computer Science and Engineering*, vol. 8, no. 12, pp. 1-5, 2021. [CrossRef] [Google Scholar] [Publisher Link]