International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 1, February 2025, pp. 755~766
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i1.pp755-766

An improved reptile search algorithm-based machine learning for sentiment analysis

Nitesh Sureja[1], Nandini M. Chaudhari[2], Jalpa Bhatt[1], Tushar Desai[1], Vruti Parikh[1], Sonia Panesar[1], Heli Sureja[3], Jahnavi Kharva[1]
[1]Department of Computer Science and Engineering, Krishna School of Emerging Technology and Applied Research, Drs. Kiran and Pallavi Patel Global University, Vadodara, India
[2]Department of Information Technology, Krishna School of Emerging Technology and Applied Research, Drs. Kiran and Pallavi Patel Global University, Vadodara, India
[3]Department of Computer Science and Engineering, Babaria Institute of Technology, Gujarat Technological University, Vadodara, India

Article Info

Article history:

Received Apr 30, 2024
Revised Aug 16, 2024
Accepted Oct 1, 2024

Keywords:

Deep learning
Machine learning
Reinforcement learning
Reptile search algorithm
Sentiment analysis
Social media
Swarm intelligence

ABSTRACT

The rapid growth of mobile technologies has transformed social media, making it crucial for expressing emotions and thoughts. When making significant decisions, businesses and governments can benefit from understanding public opinion. This information makes sentiment analysis vital for understanding public sentiment polarity. This study develops a hyper tuned deep learning model with swarm intelligence and many approaches for sentiment analysis. convolutional neural network (CNN), bidirectional encoder representations from transformers (BERT), long short-term memory (LSTM), CNN-LSTM, BERT-LSTM, and BERT-CNN are the six deep learning models of the sentiment analysis using deep learning with reinforced learning based on reptile search algorithm (SA-DLRLRSA) model. The reptile search algorithm, an enhanced swarm intelligence algorithm (SIA), optimizes deep learning model hyper parameters. Word2Vec word embedding is used to convert textual input sequences to representative embedding spaces. Pre-trained Word2Vec embedding is also used to address issue of unbalanced datasets. Experimental results demonstrate that the SA-DLRLRSA model works best with accuracies of 93.1%, 94.7%, 96.8%, 96.3%, 97.2%, and 98.3% utilizing CNN, LSTM, BERT, CNN-LSTM, BERT-CNN, and BERT-LSTM.

Corresponding Author:

Nitesh Sureja
Department of Computer Science and Engineering, Krishna School of Emerging Technology and Applied Research, Drs. Kiran and Pallavi Patel Global University
Vadodara, Gujarat, 391410, India
Email: nmsureja@gmail.com

1. INTRODUCTION

Recent interest in sentiment analysis has grown due to its many uses. Opinion mining, or sentiment analysis, uses natural language processing and deep learning to uncover subjective information and emotional states. Sentiment analysis determines if written messages are positive, negative, or neutral [1]. In recent years, social media has become vital to daily life. People express their feelings on Twitter, Meta (formerly Facebook), Instagram, and other public platforms. Therefore, social media text analysis may assist understand public opinions [2]. By reviewing customer reviews, business owners can identify product improvements. Additionally, political bodies can use sentiment analysis to create action plans [3].

Sentiment analysis (SA) is a biggest and hardest task in artificial intelligence (AI). The system uses artificial methods to recognize psychological information including attitudes, perspectives, and moods in blogs, news articles, and social media posts [4]. Social media analysis requires managing and processing enormous amounts of content. Large amounts of content were shared and generated instantly, requiring efficient content management. The content processing approach must also be considered because the contexts were not standardized like the prevalent data [5]. This work proposes a cutting-edge sentiment analysis method using swarm intelligence (SI) and deep learning. An upgraded reptile search algorithm (RSA) is used with six deep learning models: convolutional neural network (CNN), bidirectional encoder representations from transformers (BERT), long short-term memory (LSTM), CNN-LSTM, BERT-LSTM, and BERT-CNN. Reinforcement learning (RL) fine-tunes all deep learning model hyperparameters to improve RSA [6], [7]. The Word2Vec word embedding technique is used. This study addresses imbalanced datasets using data augmentation. Following paragraphs review several well-established deep learning methods integrated with swarm intelligence (SI) for sentiment analysis.

Halawani *et al.* [8] use Harris Hawks optimization and deep learning for sentiment analysis. The automated sentiment analysis in social media using Harris Hawks optimization with deep learning (ASASM-HHODL) model had 84.25%, 95.50%, and 88.75% accuracy on Sentiment140, Tweets Airline, and Tweets seminal datasets. Electroencephalography (EEG) signals were used to create an emotion recognition system in study [9] using the Shanghai Jiao Tong University (SJTU) dataset. Binary moth flame optimization (BMFO) selected features and CNN classified them. The algorithm was 95.00% accurate. Authors designed a hybrid tweet sentiment analysis algorithm in [10]. Tunicate swarm algorithm (TSA) improved scalability and processing speed in the experiment. Simulation annealing (SA) and bitwise operations are used in the hybrid HHO method to solve local optima in this work. The model had 96.37% precision. Particle swarm optimization (PSO), genetic algorithms (GA), and decision tree (DT) classifier were utilized in reference [11].

The method had 90.00% precision. GA, PSO, and decision trees were used to create a hybrid Twitter spam detection system in [12]. They create over 600 million tweets and extract attributes to detect spam in real time using uniform resource locator (URL) security. The hybrid GA-PSO-DT method is over 90.00% accurate. A deep learning model named bidirectional long short-term memory with text convolutional self-attention (BiLSTM-TCSA) was developed for short text sentiment analysis in [13]. This model uses bidirectional long short-term memory (BiLSTM), text convolutional neural network (TextCNN), and self-attention. Enhanced improved particle swarm optimization (IPSO) optimized the hyperparameters. Using a generative adversarial network (GAN), a large amount of updated text was created, improving the model's resilience. After processing, the BiLSTM model yielded global semantic insights and 94.59% accuracy on the hotel reviews dataset. Arabic Twitter sentiment analysis using PSO and deep learning (DL) was presented in study [14]. The bidirectional gated recurrent unit (BiGRU) classifier classifies attitudes. Quantum PSO (QPSO) optimizes hyperparameters.

Hybrid-flash butterfly optimization with deep learning-based sentiment analysis [15] was developed. On the Canon dataset, hybrid flower bee optimization with deep learning sentiment analysis (HFBO-DLSA) had 97.66% precision. An innovative software technique for analyzing emoji emotions was developed in [16]. Videos and images are noise-filtered first. Jiebas vocabulary was expanded by segmenting English text with emoji and internet slang. Emojis started as text. A recurrent neural network (RNN) classifies emotions as positive, extremely positive, neutral, negative, and very negative using the fuzzy butterfly optimization (FBO) algorithm. This categorization uses LSTM. The recommended sentiment analysis model outperforms current methods. Product review sentiments are categorized using the adaptive particle grey wolf optimizer with deep learning based sentiment analysis (APGWO-DLSA) in [17].

The APGWO-DLSA model obtained 94.77% accuracy on the CPAA dataset and 85.31% on the AP dataset. Alzaqebah *et al.* [18] present an improved salp swarm method (SSA) for Arabic sentiment analysis feature selection. With 80.00% accuracy, the SSA outperformed the PSO and grey wolf optimization (GWO). Mashraqi and Halawani [19] constructed dragonfly optimization with deep learning enabled Arabic tweet sentiment analysis. The term frequency-inverse document frequency (TF-IDF) model generates feature vectors. Attention-based bidirectional long short-term memory (ABLSTM) classifies sentiment. Differential flower optimization (DFO) optimizes ABLSTM hyperparameters last. On the semEval2017 dataset, the differential flower optimization with deep learning sentiment analysis and attention technique (DFODL-SAAT) model is 92.00% accurate. Log term frequency-based modified inverse class frequency (LFMI) was used to extract features in study [20]. The feature was chosen via Levy flight-based mayfly optimization. The selected data is used to build the enhanced local search whale optimization-based improved local search whale optimization with long short-term memory (ILW-LSTM) model. The ILW-LSTM method has 97% precision. In [21], a swarm intelligence algorithm called social spider algorithm (SSA) is used for the sentiment analysis within Twitter data. Decision tree, naïve Bayes, SVM, and KNN are the other classifiers used in this approach. SSA has produced very good results in comparison with other classifiers. In [22], an optimization approach with ant lion optimization (ALO) and moth flame optimization (MFO) were designed for the hate speech analysis problem. The approach achieved accuracy value was 92.1% and 90.7% with ALO, and MFO respectively.

The remaining sections of the paper are organized in the following manner: section 2 outlines the proposed sentiment analysis using deep learning with reinforced learning based on reptile search algorithm (SA-DLRLRSA) model. Section 3 provides a comprehensive overview of the performance evaluation of the proposed approach. Finally, section 4 serves as the concluding section of the entire work.

2. MATERIAL AND METHODS

This study introduces a new SA-DLRLRSA model to classify social media sentiments. Social media text is primarily transformed into useful data by SA-DLRLRSA. The SA-DLRLRSA approach reduces data-pre-processing-dependent language processing with Word2Vec word embedding.

2.1. Preparation of data

Data preparation removes unwanted and noisy data. This study includes pre-processing tasks such as, performing tokenization to convert text into a word list, streamlining SA via minimizing root proliferation, doing case conversion, performing punctuation removal from the text, performing stop words removal from the text. Neural network-based natural language processing (NLP) models are popular due to their accuracy. However, most NLP techniques perform poorly on large datasets and require word embedding for textual datasets. To improve system performance and processing speed, we used Wod2Vec word embedding. Six different deep learning models such as CNN [23], LSTM [24], BERT [25], CNN-LSTM, BERT-LSTM, and BERT-CNN are used in this study to accurately classify sentiments on social media.

2.2. Hyperparameter tuning using reptile search algorithm

An improved reptile search algorithm (RSA) adjusts these models hyperparameters to improve classification. The RSA algorithm, presented by abualigah mimics the hunting behavior of crocodiles in the wild [7]. Crocodiles may hunt on land and in water as amphibians. The basic RSA algorithm contains three steps.

2.2.1. Initialization phase

The starting solution of the RSA is produced randomly through the application of the equation $A_i^1 = LBound + ran\ \times (UBound - LBound)$. In this setting, A_i^1 represents the ith starting individual, whereas *LBound* and *UBound* refer to the lower and upper limits, respectively. Also, it denotes the current iteration count, *IT* represents the maximum iteration count.

2.2.2. Encircling phase (exploration)

Crocodiles walk high and wide during global search. Current number of iterations determines RSA search strategy. RSA walks high when *IT* is 0.25 or less. The RSA sprawl walks when *it* is less than 0.25 times the *IT* or larger than *it*. The following mathematical models describe the mechanism:

$$A_i^{t+1} = \begin{cases} A_{best}^{it} - \eta_i \times \alpha - R_i^{it} \times ran, it \leq \frac{IT}{4} \\ A_{best}^{it} \times A_{ran}^{it} \times EVS \times ran, it \leq \frac{IT}{4}\ and\ \ it > \frac{IT}{4} \end{cases} \tag{1}$$

$$\eta_i = A_{best}^{it} \times B_{it} \tag{2}$$

$$R_i = \frac{A_{best}^{it} - A_i^{it}}{A_{best}^{it} + \varepsilon} \tag{3}$$

$$EVS = 2 \times r_1 \times \left(1 - \frac{1}{IT}\right) \tag{4}$$

$$B_{it} = \beta + \frac{A_i^{it} - M(A_i^{it})}{A_{best}^{it} \times (UpperBound - LowerBound) + \varepsilon} \tag{5}$$

$$M = \frac{1}{n} \sum_{j=1}^{n} A_i \tag{6}$$

where, A_{best}^{it} represent the current best solution, α is a constant of 0.1, controls exploration rate, A_{ran}^{it} is a randomly selected individual. To avoid the denominator from being zero, the required minimal value is denoted as ε. The r_1 is a random number from -1 to 1. The constant β is set to 0.1 and, ran is a 0–1 random number. The hunting operator in the i^{th} solution is denoted as η_i which is calculated using (2). Evolutionary sense (EVS) is a random ratio between [2, -2] describe the probability of decreasing values throughout the iterations, calculated by (4). Bit corresponding to the difference between the position of the best-obtained solution and the position of the current solution, calculated by (5). M stands to the mean positions of the i^{th} solution, computed by (6).

2.2.3. Hunting phase (exploitation)

In RSA, crocodiles use two strategies for foraging: hunting coordination and cooperation. When $it < 0.75IT$ and $it \geq 0.5IT$, the RSA performs hunting coordination. When $it < IT$ and $it \geq 0.75IT$, a hunting cooperation strategy is employed by the RSA. The position updating in the hunting phase is done as (7):

$$A_i^{it+1} = \begin{cases} A_{best}^{it} - B_i \times ran, it \leq \frac{IT}{4} \; and \; it > \frac{IT}{2} \\ A_{best}^{it} \times \eta_i \times \varepsilon - R_i^{it} \times ran, it \leq IT \; and \;\; it > \frac{3IT}{4} \end{cases} \tag{7}$$

RSA generates the initial population randomly in the search space first and then chooses different search strategies depending on the number of iterations. The pseudocode for the RSA is shown in Figure 1. The RSA improves classifier efficiency with a fitness function. It assigns good-performing solutions a value greater than zero. The fitness function used in this scenario was reducing classification error rate.

$$Fitn(x_{it}) = \frac{All \; incorrectly \; classified \; instances}{All \; instances} * 100 \tag{8}$$

```
Initialize RSA parameters, create initial population randomly
While it < IT
   Calculate the Fitness of each solutions
  Find the Best solution so far
  Update the EVS using (2).
    For (i = 1 to N)
       For (j = 1 to N)
          Calculate ηi, B, R using (3), (4) and (6)
          Update Position of crocodile using (1) to (8)
      End For
it = it + 1
End While
Return the best position and fitness
```

Figure 1. Pseudocode of the RSA algorithm

2.3. Reinforcement learning

Reinforcement learning has found extensive application in various fields for problem-solving. Reinforcement learning (RL) is based on the idea that an agent changes the state of the environment by acting on it and receives a reward based on the results of the action. The two distinct kinds of reinforcement learning (RL) are value and policy-based learning. Q learning (QL) is a value-based RL method. It is a model-free, which means that the agent learns how to make the right choices in a Markovian domain [26]. The agent performs the action with the highest expected Ql value during learning. one-step Q learning is a very simple type of Q learning. In this, Ql value is changed in a single step according to the state-action pair. This work employs a one-step Q learning methodology. Each state-action pairs reward updates the Q table continuously using (9).

$$Ql(st_t, at_t) \leftarrow (1 - Lr)Ql(st_t, at_t) + Lr(rs_{t+1} + \gamma \; max_{at}Ql(st_{t+1}, at_{t+1})) \tag{9}$$

The symbols γ and Lr denote the discount factor and rate of learning, respectively. Both numbers are within the range of 0 to 1. The $Ql(st_t, at_t)$ refers to the Ql value obtained by performing action at_t in the current state st_t. On the other hand, $max_{at}Q(st_{t+1}, a_{tt+1})$ represents the highest anticipated Ql value in the Q table when executing action at_{t+1} in state st_{t+1}. It is crucial to note that an increased rate of learning (Lr) prompts the algorithm to acquire knowledge from the anticipated Ql value, whereas a decreased rate of learning prompts the algorithm to capitalize on the previous Ql value. Therefore, the rate of learning is used to strike a balance between utilizing exploiting knowledge and exploring new opportunities. Q learning pseudo-code is shown in Figure 2. Q learning randomly assigns values to the Q and reward tables. A state is then randomly chosen by the algorithm. As per lines 4 and 5, the algorithm maximizes the states future reward. This modifies the Q table, reward table, and new state.

1 Initialize Q-table and reward table with randomly
2 Chose random state st_t
3 While (Termination criteria not met)
4 Choose the best action at_t for the current state st_t from Q table
5 Execute the action and the reward rt_{t+1}
6 Get the new state st_{t+1}
7 Update Q table using (9)
8 $st_t \leftarrow st_{t+1}$
9 End While

Figure 2. Pseudocode of the Q learning algorithm

2.4. The development of the proposed SA-DLRLRSA

2.4.1. Motivation

The typical RSA technique finds solutions through exploration and exploitation. Individuals use efficient and belly-walking methods to explore new answers. Hunting operations are coordinated to find the best optimal solutions during exploitation. The algorithms capacity to change direction is limited because exploration occurs in the first half of iterations and exploitation in the second. RSAs inability to adjust iteratively makes it prone to local optima. Thus, a defined search pattern does not guarantee the optimal value. Reinforcement learning and adaptive search find the global minimum efficiently. Random opposition-based learning increases population variation to find alternative answers. These features of the RL and ROBL motivated us to use them for improving the RSA for efficient sentiment analysis.

2.4.2. The SA-DLRLRSA structure

SA-DLRLRSA uses the entire search space as its environment and every solution (individual) as an RL training agent. The Q learning algorithm switches between exploration and exploitation. The Q value of the state-action pair is updated by the Q learning algorithm using the highest fitness value and the average fitness value from earlier iterations. A table described as a reward table is used to give the punishments or incentives to the solutions (agents) based on its actions and status. The proposed SA with RL and random state learning consists of three actions that are determined by the rate of the exploration $\emptyset$: increasing the rate of the exploration, decreasing the rate of the exploration, or maintaining current rate. In the following iteration value of $\emptyset$ is adjusted considering the current highest fitness and cumulative average fitness using (10). $\emptyset^{it+1}$ indicates the rate of exploration in the following iteration, Δ represents the incremental value, and $f(X_{best}^{it})$ represents the fitness of the best position in the current iteration. The M represents the mean fitness of the fit solution (individuals) found thus far, computed using (11). Up to this point, n iterations have been done. To calculate the weighted factor for the fittest individual X_{best}^{it} at iteration it, use the formula $wf_i = e^{it/IT}$.

$$\emptyset^{it+1} = \begin{cases} \emptyset^{it} * (1+\Delta)\ if\ f(X_{best)}^{it} > M \\ \emptyset^{it} * (1-\Delta)\ if\ f(X_{best)}^{it} < M \\ \emptyset^{it} \qquad\qquad otherwise \end{cases} \tag{10}$$

$$M = \frac{1}{n}\sum_{it=1}^{n} w_i X_{best}^{it} \tag{11}$$

Here, it represents the current iteration and IT represents the total number of iterations. It is important to note that the most physically fit individuals in recent times have a greater impact on the calculation of the value of M. specifically, if the achieved fitness is higher than the average fitness, the algorithm should focus on a smaller search space and improve the acquired solutions. Alternatively, the algorithm expands its search region in order to discover novel solutions and prevent local optima. In summary, the first scenario described in (10) typically occurs when the agent achieves a higher level of fitness over the mean fitness. In the second situation, the agent's fitness starts to decline in comparison to the prior agent's experience. The SA-DLRLRSA has three states, denoted as $\mathrm{st} = \{1, -1, 0\}$ which correspond to the activities described in (12).

The reward table in this work assigns a positive value of (+1) to state $st_t = 1$ and a negative value of (-1) to all other states. If the fitness gained at iteration, it is good than the mean fitness of the last $it - 1$ iterations, then the present state st_t is equal to 1. In (13) illustrates the reward approach. Here, st_t is the state achieved by the individual (agent) at iteration *it*. Furthermore, the suggested SA-DLRLRSA method precisely adjusts the rate of learning according to the accumulated performance, as this factor greatly influences the attainment of the ideal solution. When the rate of learning is near to one, the fresh collected information significantly influences the future reward. At a low learning rate, the value of existing information surpasses that of newly acquired information. In order to optimize the outcome, the learning rate is dynamically decreased at each iteration using (14). Here, Lr_{init} and Lr_{final} represent the starting and final values of the learning rate, respectively.

$$st_t = sign(f(x^{it}) - M), sign(x) = \begin{cases} 1 & if\ x > 1 \\ -1 & if\ x < 1 \\ 0 & otherwise \end{cases} \quad (12)$$

$$Reward = \begin{cases} +1 & if st_t = 1 \\ -1 & otherwise \end{cases} \quad (13)$$

$$Lr = \frac{Lr_{init}+Lr_{final}}{2} - \frac{Lr_{init}+Lr_{final}}{2} \cdot \cos\left(\pi\left(1 - \frac{it}{IT}\right)\right) \quad (14)$$

The random opposition-based learning (ROBL) technique is incorporated into the SA-DLRLRSA algorithm to dynamically assist in avoiding the problem of being stuck in suboptimal solutions. ROBL is a technique established by [27] that use randomization to enhance the performance of optimized bee life (OBL) methods defined as: $x_{ij}^l = lb_{ij} + ub_{ij} - rand \times x_{ij}, ij = 1,2, \ldots \ldots \ldots, n$. Here, x_{ij}^l and x_{ij} denote the antithetical and initial solutions, whereas lb_j and ub_j represent the minimum and maximum limits of the variables. Figure 3 depicts the proposed SA-DLRLRSA and provides a more comprehensive explanation of how the algorithm explores the global solution.

```
Initialize RSA parameters, create initial population randomly.
Set the state st= (st1, st2, st3} and action at = (at1, at2, at3).
Initialze Q table and reward table, randomly select present state
While it < IT
   Calculate the Fitness of each solutions
Find the Best solution so far
 Update the EVS using (2).
  Update rate of exploration φ using (9)
     For (i = 1 to N)
        For (j = 1 to N)
          If rand < ∅
             If rand < 0.5
                Update crocodiles' Position using first part of (1)
              Else
                Update crocodiles'Position using second part of (1)
                End If
          Else
            If rand < 0.5
               Update crocodiles' Position using first part of (7)
            Else
               Update crocodiles' Position using second part of (7)
           End If
         End If
         Compute x_ij^l(it + 1) using ROBL
        End For
     If f(x_ij^l(it + 1)) is better than f(x_ij(it + 1))
           x_ij(it + 1) = x_ij^l(it + 1)
     End If
   Update the present state using (12)
   Compute reward using (13)
   Update Q table using (9)
   End For
it = it + 1
Update rate of learning using (14)
End While
Return the best position and fitness
```

Figure 3. Proposed SA-DLRLRSA algorithm

The algorithm starts by assessing individual fitness and finding the best solution. Next, the most advantageous action from the Q table is selected, and the exploration rate is changed using (10). The random number and exploration rate determine whether exploration or exploitation generates a new solution. The recently acquired solution is used to compute the reverse solution using ROBL. After then, the elitism mechanism determines the best solution from the reverse and new solutions. Following this, updates occur to the Q table, reward table, and present state. The rate of learning is modified after each iteration until the stopping requirements are met.

3. RESULTS AND DISCUSSION

3.1. Assessment indicators

All the assessment indicators used in this work are described in Table 1, where true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are used to calculate their values. The model is evaluated using accuracy, F1 score, precision, and recall metrics.

Table 1. Assessment indicators

Metric	Description	Formula
Accuracy	Proportion of correctly classified instances among the total instances.	$Accuracy = (TN + TP)/(TN + FP + TP + FN)$
Precision	Proportion of true positive predictions among all positive predictions.	$Precision = TP/(TP + FP)$
Recall	Proportion of true positive predictions among all actual positive instances.	$Recall = TP/(TP + FN)$
F1 Score	Harmonic mean of precision and recall, balancing between precision and recall.	$F1\ Score = (2 \times Precision * Recall)/(Precision + Recall)$

3.2. Experimental results

Table 2 shows the parameter combinations of all classifiers in this investigation. Total positive, neutral, and negative tweets are 22,937, 21,938, and 5,183, respectively. This project aims to create a Bitcoin sentiment analysis model using deep/machine learning methods. In order to enhance model effectiveness, we enhanced deep learning/machine learning parameters utilizing an enhanced reptile search algorithm (RSA). We added reinforcement learning to the RSA algorithm to boost its exploration and use of information. The models accuracy, F1 score, precision, and recall were evaluated. These metrics provide a complete assessment of the models ability to classify tweet sentiments as positive, negative, or neutral. Our study found significant differences in deep learning (DL) or machine learning (ML) algorithm efficacy. Table 3 compares models performance factors.

Table 2. Parameter setting of the models

Parameters	Values					
	CNN	LSTM	BERT	CNN-LSTM	BERT-CNN	BERT-LSTM
Act. function	Softmax	Softmax	Softmax	Softmax	Softmax	Softmax
Batch size	128	128	128	128	128	128
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Epochs	10	10	10	10	10	10
Learning rate	----	----	0.001	-------		

Table 3. Results attained by the models

Models	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
CNN	93.10	93.96	92.10	93.02
LSTM	94.97	96.19	95.85	96.02
BERT	96.84	97.44	97.18	97.31
CNN-LSTM	96.38	96.44	96.22	96.33
BERT-CNN	97.22	97.12	96.90	97.01
BERT-LSTM	98.32	97.28	97.18	97.23

Figure 4 shows the graphs tweet numerically. Figure 5 shows the results graphically. Figure 6 shows the CNN classifier achieved 96.5% accuracy, 93.9% precision, 92.1% recall, and 93.0% F1 score. Figure 7 shows that the LSTM classifier had 94.9% accuracy, 96.1% precision, 95.8% recall, and 96.0% F1 score. Figure 8 shows that the BERT classifier had 96.8% accuracy, 97.4% precision, 97.1% recall, and 97.3% F1 score. The CNN-LSTM classifier in Figure 9 had 96.3% accuracy, 96.4% precision, 96.9% recall, and 97.0% F1 score. Figure 10 shows the BERT-CNN classifiers 97.2% accuracy, 97.1% precision, 92.1% recall, and 93.0% F1 score. Figure 11 shows that the BERT-LSTM classifier outperformed all other models. The model had 98.3% accuracy, 97.2% precision, 97.1% recall, and 97.2% F1 score. BERT-LSTM was resilient and effective at assessing Bitcoin tweet sentiment. Our research also compares our findings to earlier studies [16]–[18], [21]. Our work shows that deep learning/machine learning improves classification accuracy through real-time tweet detection and analysis. It is also important to highlight constraints like dataset size and biases that may restrict outcomes of the research. This may require intelligent hyperparameter adjusting, using domain-specific characteristics based on expert knowledge, or adding external data to augment the dataset. We can improve sentiment analysis by incorporating these variables and improving categorization algorithms.

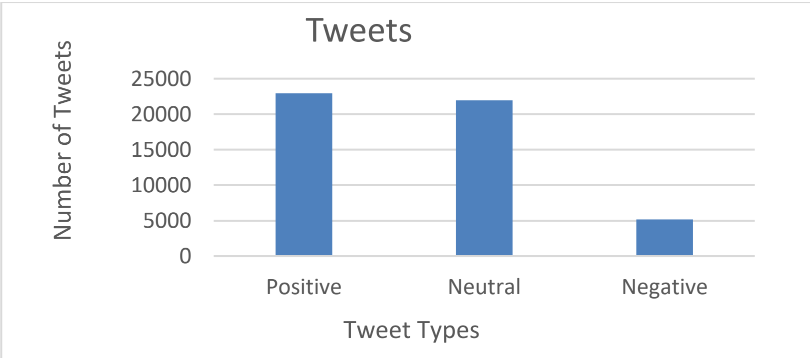


Figure 4. Number of Tweets

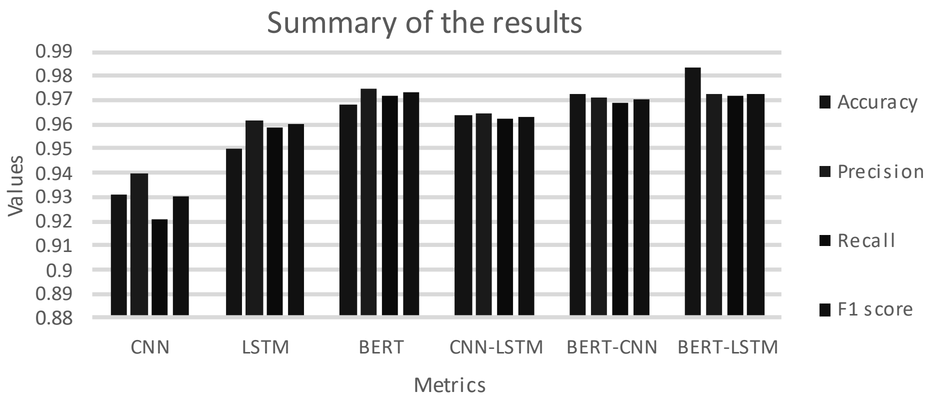


Figure 5. Results obtained

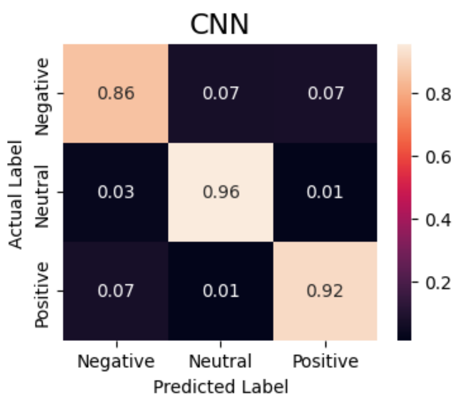


Figure 6. CNN confusion matrix

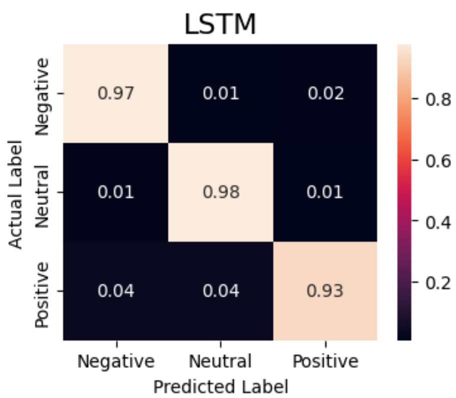


Figure 7. LSTM confusion matrix

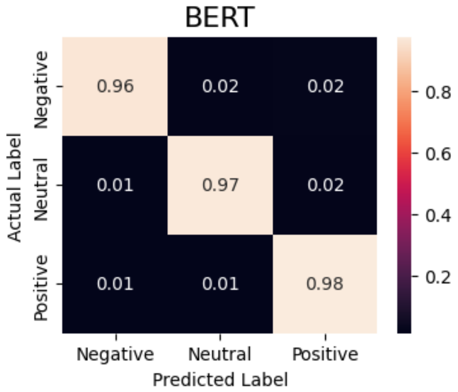


Figure 8. BERT confusion matrix

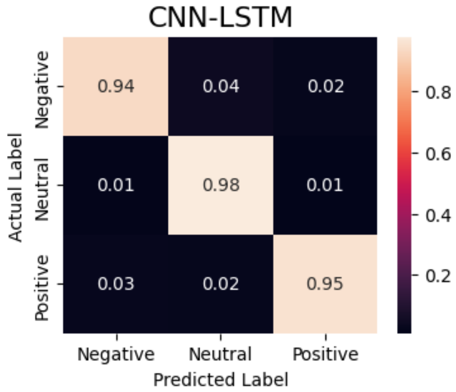


Figure 9. CNN-LSTM confusion matrix

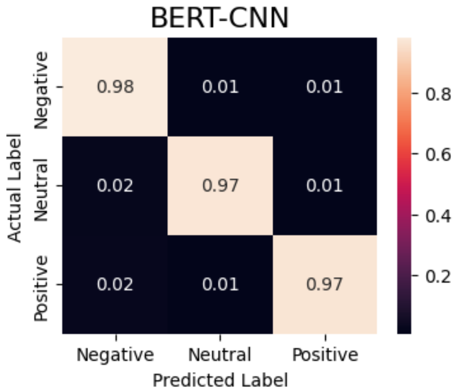


Figure 10. BERT-CNN confusion matrix

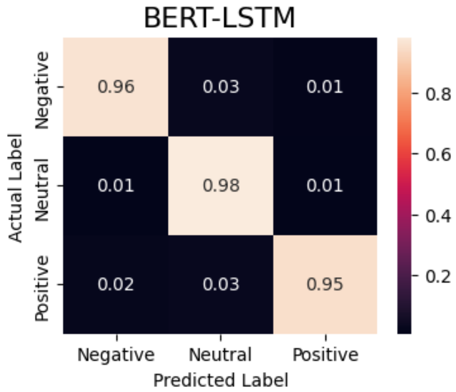


Figure 11. BERT-LSTM confusion matrix

4. CONCLUSION

This work presents the development of the SA-DLRLRSA algorithm for sentiment classification of Bitcoin tweets. The objective of the SA-DLRLRSA technique is to develop an automated artificial intelligence model that accurately classifies the tweets as positive, negative, or neutral in terms of their sentiment towards Bitcoins. The SA-DLRLRSA technique consists of four stages: data preparation, preprocessing, sentiment classification based on deep learning or machine learning, and hyperparameter tuning based on improved RSA. The RSA technique is employed for hyperparameter tunning in order to enhance the results of the DL or ML algorithms. The effectiveness of the SA-DLRLRSA approach is confirmed by testing it on the Bitcoin tweets dataset obtained from the Kaggle repository. The experimental results showed that the SA-DLRLRSA method outperformed other present algorithms in several measures. This analysis offers a valuable insight into the publics emotions towards Bitcoins on Twitter, enabling a better comprehension and evaluation of its influence and perception among users. In the future, the presented model has the potential to be expanded for classifying views related to various topics. Furthermore, there are several additional strategies that can be employed to enhance the performance of the suggested SA-DLRLRSA model.

REFERENCES

[1] C. A. Iglesias and A. Moreno, "Sentiment analysis for social media," *Applied Sciences*, vol. 9, no. 23, pp. 1–4, Nov. 2019, doi: 10.3390/app9235037.

[2] Q. Tul *et al.*, "Sentiment analysis using deep learning techniques: a review," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, pp. 424–433, Jan. 2017, doi: 10.14569/IJACSA.2017.080657.

[3] D. Li, R. Rzepka, M. Ptaszynski, and K. Araki, "HEMOS: a novel deep learning-based fine-grained humor detecting method for sentiment analysis of social media," *Information Processing and Management*, vol. 57, no. 6, Nov. 2020, doi: 10.1016/j.ipm.2020.102290.

[4] M. Sinan *et al.*, "Analysis of the mathematical model of cutaneous leishmaniasis disease," *Alexandria Engineering Journal*, vol. 72, pp. 117–134, Jun. 2023, doi: 10.1016/j.aej.2023.03.065.

[5] A. R. Pathak, B. Agarwal, M. Pandey, and S. Rautaray, "Application of deep learning approaches for sentiment analysis," in *Algorithms for Intelligent Systems*, 2020, pp. 1–31.
[6] D. Wu, S. Wang, Q. Liu, L. Abualigah, and H. Jia, "An improved teaching-learning-based optimization algorithm with reinforcement learning strategy for solving optimization problems," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–24, Mar. 2022, doi: 10.1155/2022/1535957.
[7] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer," *Expert Systems with Applications*, vol. 191, Apr. 2022, doi: 10.1016/j.eswa.2021.116158.
[8] H. T. Halawani, A. M. Mashraqi, S. K. Badr, and S. Alkhalaf, "Automated sentiment analysis in social media using Harris Hawks optimisation and deep learning techniques," *Alexandria Engineering Journal*, vol. 80, pp. 433–443, Oct. 2023, doi: 10.1016/j.aej.2023.08.062.
[9] T. A. Tuib, B. H. Saoudi, Y. M. Hussein, T. H. Mandeel, and F. T. Al-Dhief, "Convolutional neural network with binary moth flame optimization for emotion detection in electroencephalogram," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, pp. 1172–1178, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp1172-1178.
[10] R. Seth and A. Sharaff, "Sentiment data analysis for detecting social sense after COVID-19 using hybrid optimization method," *SN Computer Science*, vol. 4, no. 5, Jul. 2023, doi: 10.1007/s42979-023-02017-3.
[11] S. M. Nagarajan and U. D. Gandhi, "Classifying streaming of Twitter data based on sentiment analysis using hybridization," *Neural Computing and Applications*, vol. 31, no. 5, pp. 1425–1433, May 2019, doi: 10.1007/s00521-018-3476-3.
[12] N. S. Murugan and G. Usha Devi, "Detecting streaming of Twitter spam using hybrid method," *Wireless Personal Communications*, vol. 103, no. 2, pp. 1353–1374, Nov. 2018, doi: 10.1007/s11277-018-5513-z.
[13] Y. Yue, Y. Peng, and D. Wang, "Deep learning short text sentiment analysis based on improved particle swarm optimization," *Electronics*, vol. 12, no. 19, pp. 1–23, Oct. 2023, doi: 10.3390/electronics12194119.
[14] B. B. Al-onazi *et al.*, "Quantum particle swarm optimization with deep learning-based Arabic tweets sentiment analysis," *Computers, Materials {\and} Continua*, vol. 75, no. 2, pp. 2575–2591, Jan. 2023, doi: 10.32604/cmc.2023.033531.
[15] P. Manjula, M. Maragatharajan, P. Rajput, S. P. R. Kamala, N. Kopperundevi, and S. N. Sangeethaa, "Sentiment analysis for online product reviews and recommendation using deep learning based optimization algorithm," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 9, pp. 3629–3640, Nov. 2023, doi: 10.17762/ijritcc.v11i9.9585.
[16] J. Venkataraman and L. Mohandoss, "FBO-RNN: fuzzy butterfly optimization-based RNN-LSTM for extracting sentiments from Twitter emoji database," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 12, May 2023, doi: 10.1002/cpe.7683.
[17] D. Elangovan and V. Subedha, "Adaptive particle grey wolf optimizer with deep learning-based sentiment analysis on online product reviews," *Engineering, Technology {\and} Applied Science Research*, vol. 13, no. 3, pp. 10989–10993, Jun. 2023, doi: 10.48084/etasr.5787.
[18] A. Alzaqebah, B. Smadi, and B. H. Hammo, "Arabic sentiment analysis based on salp swarm algorithm with S-shaped transfer functions," in *2020 11th International Conference on Information and Communication Systems (ICICS)*, Apr. 2020, pp. 179–184, doi: 10.1109/ICICS49469.2020.239507.
[19] A. M. Mashraqi and H. T. Halawani, "Dragonfly optimization with deep learning enabled sentiment analysis for Arabic tweets," *Computer Systems Science and Engineering*, vol. 46, no. 2, pp. 2555–2570, Jan. 2023, doi: 10.32604/csse.2023.031246.
[20] L. R. Krosuri and R. S. Aravapalli, "Feature level fine grained sentiment analysis using boosted long short-term memory with improvised local search whale optimization," *PeerJ Computer Science*, vol. 9, pp. 1–24, Apr. 2023, doi: 10.7717/peerj-cs.1336.
[21] Baydogan and B. Alatas, "Sentiment analysis in social networks using social spider optimization algorithm," *Tehnicki Vjesnik - Technical Gazette*, vol. 28, no. 6, pp. 1943–1951, Dec. 2021, doi: 10.17559/TV-20200614172445.
[22] C. Baydogan and B. Alatas, "Metaheuristic ant lion and moth flame optimization-based novel approach for automatic detection of hate speech in online social networks," *IEEE Access*, vol. 9, pp. 110047–110062, Jan. 2021, doi: 10.1109/ACCESS.2021.3102277.
[23] U. D. Gandhi, P. Malarvizhi Kumar, G. Chandra Babu, and G. Karthick, "Sentiment analysis on Twitter data by using convolutional neural network (CNN) and long short term memory (LSTM)," *Wireless Personal Communications*, pp. 1–10, May 2021, doi: 10.1007/s11277-021-08580-3.
[24] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi: 10.1162/neco_a_01199.
[25] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Jan. 1997, doi: 10.1109/78.650093.
[26] B. M. Kayhan and G. Yildiz, "Reinforcement learning applications to machine scheduling problems: a comprehensive literature review," *Journal of Intelligent Manufacturing*, vol. 34, no. 3, pp. 905–929, Mar. 2023, doi: 10.1007/s10845-021-01847-3.
[27] W. Long, J. Jiao, X. Liang, S. Cai, and M. Xu, "A random opposition-based learning grey wolf optimizer," *IEEE Access*, vol. 7, pp. 113810–113825, Jan. 2019, doi: 10.1109/ACCESS.2019.2934994.

BIOGRAPHIES OF AUTHORS

Nitesh Sureja is currently employed with KPGU University, Vadodara, and Gujarat, India. He is working as the director at the Krishna School of Emerging Technology and Applied Research, one of the constituent institutes of KPGU. His research interests include machine learning, data science, artificial intelligence, and algorithms inspired by nature. He has been a teacher for more than 23 years. Publications with SCOPUS and Google indexes have published his study findings. He has mentored two Ph.D. candidates and is currently mentoring five Ph.D. candidates. He can be contacted at nmsureja@gmail.com.

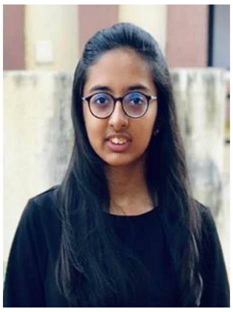
Nandini M. Chaudhari is recently working as professor and head in the IT Department, Krishna School of Emerging Technology and Applied Research, KPGU University, Vadodara, Gujarat. She has more than 30 years of experience. She is currently guiding six research scholars under the Faculty of Science and Technology at North Maharashtra University and KPGU. Her 17 papers have been published in journals indexed in SCOPUS, WOS, UGC-listed journals, and other reputed journals. Her research interests include image and video processing, AI/ML, IoT, and quantum computing. She has published and filed two patents. She is a member of the Board of Studies in computer science and engineering at NMU and KPGU. She can be contacted at email: nandini113@gmail.com.

Jalpa Bhatt is a seasoned professional holding a master degree in computer science and engineering with a rich background encompassing eight years of experience in both the realms of education and industry. As an expert in web recommendation, she has honed her skills to deliver personalized and effective solutions, contributing to enhanced user experiences in the digital landscape. The dynamic realm of IoT has been a significant area of expertise for her. Over the years, she has harnessed the power of IoT to create innovative and interconnected solutions, driving efficiency and connectivity in various applications. She can be contacted at email: jalpabhattpj@gmail.com.

Tushar Desai is currently working as an assistant professor in the Computer Science and Engineering Department, Krishna School of Emerging Technology and Applied Research, Drs. Kiran and Pallavi Patel Global University, Vadodara, Gujarat (KPGU). He holds a master degree in computer science and engineering and has experience spanning over a decade in both education and industry. With a decade of hands-on experience, He has honed his skills in cloud computing, where he possesses a profound understanding of the latest trends and technologies. He can be contacted at email: er.tushar.it@gmail.com.

Vruti Parikh is currently working as an assistant professor with Drs. Kiran and Pallavi Patel Global University (KPGU), Vadodara, Gujarat, India. Her current research interests include data deduplication, data optimization, and data science. Her research pursuits reflect her dedication to exploring innovative solutions in these areas, contributing to advancements in the field of data management and analytics. She is assisting students with various training and placement activities. She worked on various live projects during her industrial experience. She can be contacted at email: vrutiparikh.cse.kset@kpgu.ac.in.

Sonia Panesar is an assistant professor in the CSE department at KPGU, with five years of experience at the institution and a total of nine years in the field. She completed her B.Tech. in CSE from DAVIET, Jalandhar, and her M.Tech. in CE from Parul University in 2019. Currently, she is pursuing a Ph.D. in CSE with a specialization in machine learning and deep learning with medical databases from Parul University. She can be contacted at email: soniafpanesar@gmail.com.



Heli Sureja is currently a student of B.E. computer science and engineering at Department Computer Science and Engineering, Babaria Institute of Technology-Vadodara, Gujarat Technological University, Ahmedabad, Gujarat, India. She has published one research paper in a Scopus indexed journal. She can be contacted at email: helisureja17@gmail.com.

Jahnavi Kharva is currently working at the Department of Computer Science and Engineering, Krishna School of Emerging Technology and Applied Research, Drs. Kiran and Pallavi Patel Global University, Vadodara, Gujarat (KPGU). She has completed her B.E. in computer engineering from Sardar Vallabhbhai Patel Institute of Technology, Vasad. Her current interests are in data analysis, statistical modeling, Python, machine learning, and data visualization. She can be contacted at email: kharvajahnavi@gmail.com.