International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 10 | Oct 2021 www.irjet.net p-ISSN: 2395-0072

CHAMELEON SWARM OPTIMISATION WITH MACHINE LEARNING BASED SENTIMENT ANALYSIS ON SARCASM DETECTION AND CLASSIFICATION MODEL

Abel Sridharan[1]

[1]Senior Digital Engineering Leader, Computer Science and Data Science, Chennai, Tamil Nadu, India

---***---

Abstract - *Sentimental analysis and sarcasm detection are considered challenging issues in social media and Internet owing to the fact that people express their negative thought, hate, and thoughts by means of positive vocabulary. Sarcasm detection becomes essential and gained significant interest due to the fact that the online content is composed of positive words implies negative thoughts. Automated sarcasm detection and classification using statistical approaches and machine learning (ML) models remains a hot research topic among research communities as well as business people. This study develops a novel chameleon swarm optimization (CSO) with machine learning based SA on sarcasm detection and classification (CSOML-SASC) model. The proposed CSOML-SASC technique involves pre-processing, feature extraction, classification, and parameter optimization. In addition, the CSOML-SASC technique involves TF-IDF based feature extraction model. Besides, CSO with weighted kernel extreme learning machine (WKELM)) based classification model has been derived in which the parameter optimization process takes place to improve the overall performance. In order to showcase the enhanced performance of the CSOML-SASC technique, a series of simulations take place on benchmark datasets and the results demonstrated the better performance of the CSOML-SASC technique over the other recent approaches.*

Key Words: Sarcasm detection, Sentiment analysis, Machine learning, Parameter tuning, Chameleon swarm optimization algorithm

1.INTRODUCTION

Nowadays, social media platforms like Twitter and Facebook are considered novel ways of knowledge exchange, collaboration, and social interaction. Social media platforms enable users to exchange comments, ideas posts, updates, and reviews when sharing their common interests [1]. Because of the excessive use of social networking sites, it is essential to employ related analysis tasks manually. This task consists of offensive language detection, sarcasm detection, irony detection, sentimental analysis (SA), and author profiling. SA is expensive since it enables users to capture a summary of social networking users' opinions towards certain services, topics, or products. [2] The SA approach represents the usage of Machine Learning (ML) and Natural Language Processing (NLP) to extract and identify subjective data in this article. It is considered a common phenomenon in social networking sites [3], i.e., complex to analyze automatically for humans. Even though it has a considerable impact on sentiments, it is neglected in social networking analysis since it is considered very difficult to manage. Hence, sarcasm is still considering a challenging problem in SA approach [4].

Sarcasm means several emotions and positive words are posted on Twitter which represents slang, undesirable, or negative features [5]. Sarcasm has the capacity to convey opinions or negative emotions via aggravating/positive words. E.g., 'I love working here for nothing' is a sarcastic word in which 'love' is utilized as mockery or irony for criticizing the workplace. Different from humans who can understand its meaning easily, the usage of positive words making it extremely difficult for ML methods for understanding the figurative nature of the sarcasm in text. Consequently, sarcasm could change the polarity of tweets from negative to positive words [6]. Sarcasm detection is considered a tedious process in SA method because it is hard to define the sharpness, intensity, and pitch of voice in textual information that frequently assist to understand sarcasm [7]. The importance of sarcasm detection for SA method and its challenges makes it an increasing research problem. Various methods for sarcasm detection were introduced which integrate machine learning, statistical, and rule-based methods however mostly they exploit simple datasets. But this method cannot perceive the figurative meaning of the word. Moreover, this approach requires handcrafted features and isn't able to understand the pattern in passive voice sentences [8]. But, in place of using handcrafted features, methods integrating deep neural networks (DNN) automatically learn the imperative features.

This study introduces an efficient chameleon swarm optimization (CSO) with machine learning based SA on sarcasm detection and classification (CSOML-SASC) model. The proposed CSOML-SASC technique involves TF-IDF based feature extraction model to generate feature vectors. Moreover, the CSO with weighted kernel extreme learning machine (WKELM)) based classifier is designed in which the parameter optimization process is carried out for improving the overall classification results. For examining the betterment of the CSOML-SASC technique, a wide range of experiments were carried out on benchmark datasets and the results are inspected under varying aspects.

2. LITERATURE REVIEW

Mandal and Mahto [9] proposed a DNN method which leverages the benefits of CNN and LSTM models. Then, This CNN-LSTM-NN method using word embedding is trained 21,709 on word vector encoding of headline news for determining either headline is genuine/sarcastic. Then, we employed this NN for a corpus of 5000 test samples of sarcastic and real news headlines. In Mohammed et al. [10], effective methods in short-text classification are verified to detect sarcasm in the Arabic news headline originally. The dataset is utilized to train and test these distinct features was automatically gathered by scrapping 2 distinct websites, non-sarcastic and sarcastic. Also, Comprehensive result for all the models is characterized, according to the distinct performance metrics, like F1 score, accuracy, recall, and precision.

Bharti et al. [11] design an architecture for the detection of sarcasm in Hindi tweets with the help of online news. In this work, the online news is regarded as the context of a provided tweet at the time of sarcasm detection. Farha and Magdy [12] estimate the performances of 24 of these methods on Arabic sarcasm and sentiment detection. The result shows that the method achieves outstanding performances are trained on Arabic data, includes dialectal Arabic, and utilizes a large amount of parameters, like the currently published MARBERT. But we observed that AraELECTRA is the topmost performing method while being very effective in its computation costs.

In Nayel et al. [13], an approach based supervised ML method named SVM model was utilized. The presented method was estimated by ArSarcasm-v2 datasets. The performances of the presented method were compared to another method presented to sarcasm detection and SA distributed tasks. Chudi-Iwueze and Afli [14] focusses on the effects of distinct feature encoding methods employed to text for feature extraction for ML methods. Also, the DL method employed and the outcomes are compared. Previous to feature extraction, data pre-processing algorithms namely punctuation, tokenization, and removal of stop-words are used by the authors in this study including text analysis. These pre-processing methods are extensively accepted and employed. Distinct feature extraction approaches such as Term Frequency-Inverse Document Frequency, word embedding, and Count Vectorizer are executed in this work. The SVM, NB, and LR are the conventional ML method utilized in this work.

Shrikhande et al. [15] construct sarcasm detector with NN and attempt to understand how a computer learns the pattern of sarcasm. The input to this project comprises of sequence i.e., labelled non-sarcastic/sarcastic. This sequence comes from 2 distinct datasets comprising social media commentary and news headlines. The classifier is estimated on their precisions. This method achieves extremely and able to consistently classify sarcastic or non-sarcastic sentences. Jamil et al. [16] introduce a hybrid model in which the CNN method is utilized for feature extraction whereas the LSTM is tested and trained on this feature. For accuracy analysis, various ML methods like RF, SVM, further DT, and tree classifier are utilized. The performances of the presented method and ML models are examined by the TF-IDF, BoW approaches, and global vector for word representation.

3. THE PROPOSED CSOML-SASC TECHNIQUE

In this study, a novel CSOML-SASC technique is derived for SA on sarcasm detection and classification. The proposed CSOML-SASC technique involves pre-processing, TF-IDF based feature extraction, WKELM based classification, and CSO based parameter optimization. The design of CSO algorithm for tuning the weight and bias values of the WKELM model helps to accomplish improved classification performance. Fig. 1 showcases the overall process of block diagram of CSOML-SASC model.

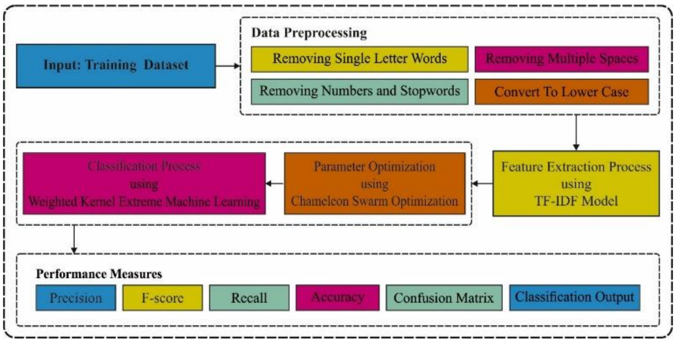

Fig. 1. Block diagram of CSOML-SASC model

3.1 Pre-processing

In the primary stage, the data has been pre-processed for transforming to compatible format. The distinct sub procedures contained from data pre-processed are:

- Extraction of single letter words
- Elimination of several spaces
- Extraction of punctuation marks
- Elimination of numerals
- Elimination of stop words and
- Convert uppercase letters into lowercase.

3.2 TF-IDF based Feature Extraction

TF-IDF is one of the commonly utilized feature extraction approaches in text analysis. Among the two essential tasks of index and weight to text analysis, TF-IDF manages the weighting. It determines the weight of provided term t from the provided document D. The TF-IDF has developed in TF and IDF that are distinct terms and is computed as

$$TF(t) = \frac{t_D}{N_D} \tag{1}$$

$$IDF(t) = \frac{\log d}{dt} \quad (2)$$

where t_D, d and dt imply the entire amount of t presence in document D, entire amount of documents, and the amount of documents which comprise term t.

The weight of all terms utilizing the TF-IDF has been computed with [17]:

$$W_{t,d} = TF_{t,d}\left(\frac{t_D}{d_{f,t}}\right) \quad (3)$$

where $TF_{t,d}$ and $d_{f,t}$, refers the frequency of term t from the document d and amount of documents that comprise t.

3.3 WKELM based Classification

At this stage, the derived feature vectors are passed into the WKELM model to detect and classify sarcasm. ELM is commonly employed for classification process. The parameters a_i and b_i are arbitrarily created. It does not modify under the entire technique. The hidden layer has been defined after the input parameters have been formed. According to the input parameters and hidden layers, it is developing the outcome with the linear analytic solutions. Fig. 2 illustrates the framework of ELM model. The last aim of ELM is for attaining the least trained error with lowest norm of resultant weight. It can be written as:

$$\underset{a_i,b_i,\beta}{arg\min} \|H(a_1,\dots,a_L;b_1,\dots,b_L)\beta - Y\|^2. \quad (4)$$

According to optimized concept Eq. (4) has been expressed as follows:

$$\min_{\beta}\frac{1}{2}\|\beta\|_2^2 + C\frac{1}{2}\xi_i^2 s.t.\, h(x)\beta = y_i^T - \xi_i^T, i = 1,\dots,N, \quad (5)$$

where $h(x) = [G(a_1,b_1,x),\dots,G(a_L,b_L,x)]$, ξ_i signifies the trained error, and C refers the regularizing parameter.

Based on Lagrange multiplier scheme and KarushKuhnTucker (KKT) optimized condition [18], trained the ELM has been corresponding for resolving the subsequent dual optimize issue:

$$\min_{(\beta,\alpha,\xi_i)} L_{ELM} = \frac{1}{2}\|\beta\|_F^2 + C\frac{1}{2}\sum_{i=1}^{N}\|\xi_i\|_F^2 - \sum_{i=1}^{N}\sum_{j=1}^{M}\alpha_{i,j}\left(h^T(x_i)\beta_j - y_{i,j} + \xi_{i,j}\right), \quad (6)$$

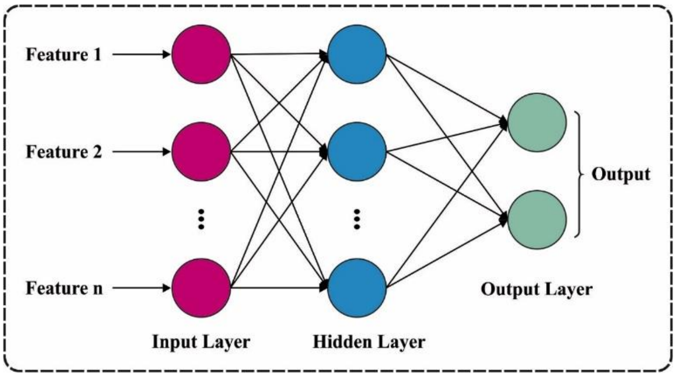

Fig. 2. ELM structure

where β_j indicates the column vector of matrix β and $\alpha_{i,j}$ represents the Lagrange multiplier. In the KKT statement, it can more develop:

$$\frac{\partial L_{ELM}}{\partial \beta_j} = 0 \to \beta = H \times \alpha \quad (7)$$

$$\frac{\partial L_{ELM}}{\partial \varepsilon_i} = 0 \to \alpha_i = C\varepsilon_i, i = 1,\dots,N \quad (8)$$

$$\frac{\partial L_{ELM}}{\partial \alpha_i} = 0 \to H^T\beta = y_i^T - \xi_i^T = 0, i = 1,\dots,N \quad (9)$$

According to Eqs. (9)-(11), the resultant weight, β, has stated as

$$\beta = H^T\left(\frac{I}{C} + HH^T\right)^{-1} Y. \quad (10)$$

Next to achieving the resultant weight β, the outcome of ELM has reached as:

$$f(x) = h(x)\beta. \quad (11)$$

The standard ELM doesn't take the imbalance issue as to account, but the weighted ELM is planned for addressing it. In this case, 2 weighting techniques are presented:

$$W = \frac{1}{t_k}, \quad (12)$$

where t_k stands for the entire amount of instances fitting to kth class. Afterward implementing weighting scheme 1, it can attain a balanced ratio amongst the minority as well as majority

$$W = \begin{cases} \frac{0.618}{t_k} & if\ (t_k > t_{avg}) \\ \frac{1}{t_k} & if\ (t_k \leq t_{avg}) \end{cases}, \quad (13)$$

where t_{avg} implies the average amount of instances in every class. When the amount of t_k is lower the average, related to ELM, the optimized procedure of weighted ELM is written as:

$$\min_{\beta} \frac{1}{2} \|\beta\|_2^2 + C_2^1 W \xi_i^2, \quad (14)$$

$$s.t.\, h(x)\beta = y_i^T - \xi_i^T, i = 1, \dots, N.$$

To the multiclass-weighted kernel ELM, it can determine a diagonal matrix, W that is linked with trained instance x. The resultant weight, β, is written as:

$$\beta = H^T \left(\frac{I}{C} + HWH^T\right)^{-1} WY. \quad (15)$$

To provide a novel instance, x, the resultant function of weighted ELM classifier has attained in $f(x) = h(x)\beta$, that is:

$$f(x) = h(x)\beta = h(x)H^T \left(\frac{I}{C} + WHH^T\right)^{-1} WY. \quad (16)$$

Related to SVM kernel techniques, the kernel trick is utilized in Eq. (18), where the kernel function changes the inner product $h(x)H^T$ and HH^T. The kernel trick type of weighted ELM is the weighted kernel ELM. Therefore, the $N \times N$ version of kernel ELM has formulated as:

$$f(x) = \begin{pmatrix} k(x, x_1) \\ \vdots \\ k(x, x_N) \end{pmatrix}^T \left(\frac{I}{C} + WK(x_i, x_j)\right) WY, \quad (17)$$

where $h(x)H^T = \begin{pmatrix} k(x, x_1) \\ \vdots \\ k(x, x_N) \end{pmatrix}^T$ and $HH^T = K(x_i, x_j)$.

So, the weighted kernel ELM offers a unified solution to network with distinct feature mapping and, simultaneously, strengthens the influence of minority class instances with more a weighted matrix.

3.4 CSO based Parameter Optimization

For optimally determining the hyperparameter involved in the CSO algorithm, the CSO algorithm is employed to it and thereby improves the overall classification performance. The CSO [19] algorithm is a metaheuristic technique and it follows initialization of the population to the resolve of the optimized procedure. Assume that the entire amount of population is C and obtainable under the search space of D. A primary population has been created from the dimensional together with arbitrary initialized under the search space has provided as:

$$a^i = L_j + rand \times (U_j - L_j) \quad (18)$$

A primary vector of ith chameleon has represented as a^i. The lower as well as upper limits of search region are provided as L_j and U_j from the jth dimensional correspondingly. $rand$ refers to the arbitrarily created number that falls in the range of [0-1].

The improved capability of chameleons for searching under the search space has been written as:

$$\rho = \delta \exp^{(-\alpha t/R)} \quad (19)$$

At this point, ρ signifies the parameter utilized from the iteration and decreases with the improving iterations. δ, α, and β demonstrate the existing parameters employed for managing the exploration as well as exploitation phases. The rotating centred coordinate utilized to upgrade the place of chameleons under the search space has provided as:

$$arand_r^i = m \times ac_r^i \quad (20)$$

$arand_r^i$ implies the rotating centered coordinate of chameleon. m refers the utilized for denoting the rotation matrix and ac_r^i has utilized for denoting the center coordinates at rth iteration. The inertia weight of iterations is provided as:

$$W = (1 - r/R)^{(\lambda\sqrt{r/R})} \quad (21)$$

At this point, W refers the weight of inertias, λ represents the arbitrary number utilized for controlling the exploitation ability [20]. The value of λ is equivalent to one. The acceleration rate of chameleon has provided as:

$$y = 2590 \times \left(1 - \exp^{-\log(r)}\right) \quad (22)$$

where y has utilized to define the acceleration of chameleon. It is observed that the CSO initialization the optimized procedure and the chameleon places are upgraded utilizing the formulas

$$a_{r+1}^{ij} = \begin{cases} a_r^{ij} + p_1(P_r^{ij} - G_r^j)rand_1 + p_2(G_r^j - a_r^{ij})rand_2 & rand_i \geq P \\ a_r^{ij} + \rho(U^j - L^j)rand_3 + L_b^j sgn(rand - 0.5) & rand_j < P \end{cases} \quad (23)$$

$$a_{r+1}^i = arand_r^i + \overline{a}_r^i \quad (24)$$

$$a_{r+1}^i = \overline{a}_r^i + ((v_r^{ij})^2 - (v_{r-1}^{ij})^2)/(2y) \quad (25)$$

At this point, G_r^j refers the global optimum place of chameleons and v_r^{ij} represents the novel velocity of rth chameleon. If some of chameleon go exterior of search space afterward it is sent back to constraint that is defined already. The fitness function (FF) has been evaluated inside all the iterations for predicting the chameleon with optimum fitness. Therefore, the FF has been utilized for finding the optimum chameleon that catches the prey previously. These can be frequent until it fulfils the entire iteration cycles.

4. EXPERIMENTAL VALIDATION

This section investigates the results analysis of the CSOML-SASC technique on the benchmark Kaggle News headlines dataset [21]. Fig. 3 showcases the confusion matrices generated by the CSOML-SASC technique under three distinct runs. Under run-1, the CSOML-SASC technique has identified 14237 instances into sarcastic and 12662 instances into non-sarcastic. Also, under run-2, the CSOML-SASC manner has identified 14281 instances into sarcastic and 12750 instances into non-sarcastic. Moreover, under run-3, the CSOML-SASC approach has identified 14299 instances into sarcastic and 12802 instances into non-sarcastic.

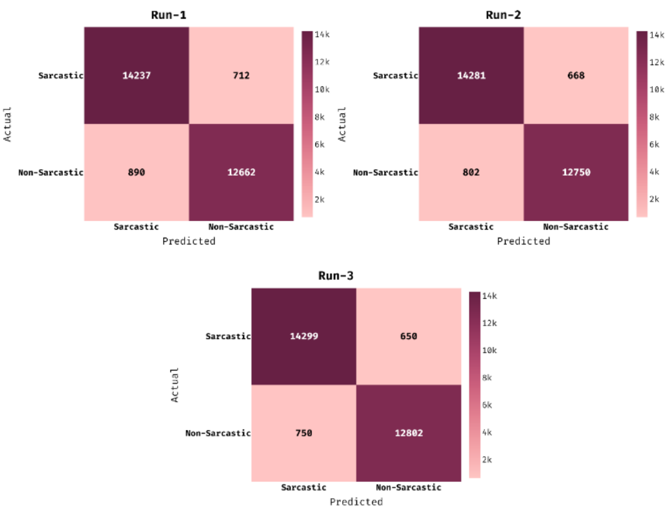

Fig. 3. Confusion matrix analysis of CSOML-SASC model

Table 1 and Fig. 4 offer the overall classification results analysis of the CSOML-SASC technique on the test dataset. On the applied test run-1, the CSOML-SASC technique has gained a precision of 0.9412, recall of 0.9524, accuracy of 0.9438, and F-score of 0.9467. Likewise, on the applied test run-2, the CSOML-SASC method has reached a precision of 0.9468, recall of 0.9553, accuracy of 0.9484, and F-score of 0.9511. Similarly, on the applied test run-3, the CSOML-SASC algorithm has attained a precision of 0.9502, recall of 0.9565, accuracy of 0.9509, and F-score of 0.9533.

Table 1 Result analysis of CSOML-SASC model with different measures

No. of Runs	Precision	Recall	Accuracy	F-Score
Run-1	0.9412	0.9524	0.9438	0.9467
Run-2	0.9468	0.9553	0.9484	0.9511
Run-3	0.9502	0.9565	0.9509	0.9533
Average	**0.9461**	**0.9547**	**0.9477**	**0.9504**

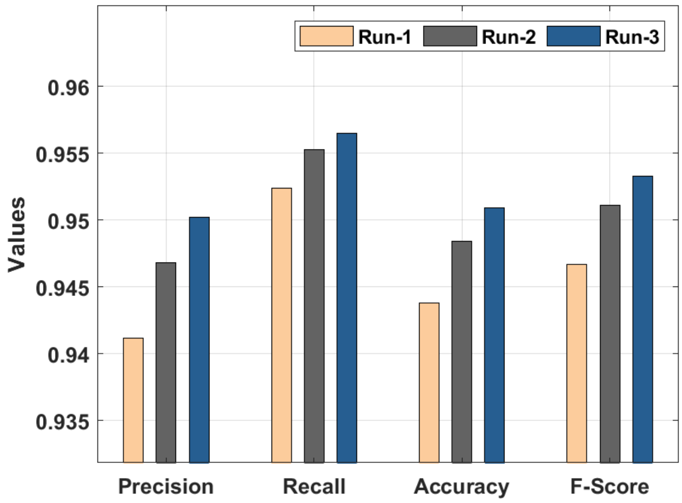

Fig. 4. Result analysis of CSOML-SASC technique with varying measures

Fig. 5 provides a detailed average classification results analysis of the CSOML-SASC technique on the applied test dataset. The figure portrayed that the CSOML-SASC technique has accomplished maximum outcome with the average precision of 0.9461, recall of 0.9547, accuracy of 0.9477, and F-score of 0.9504.

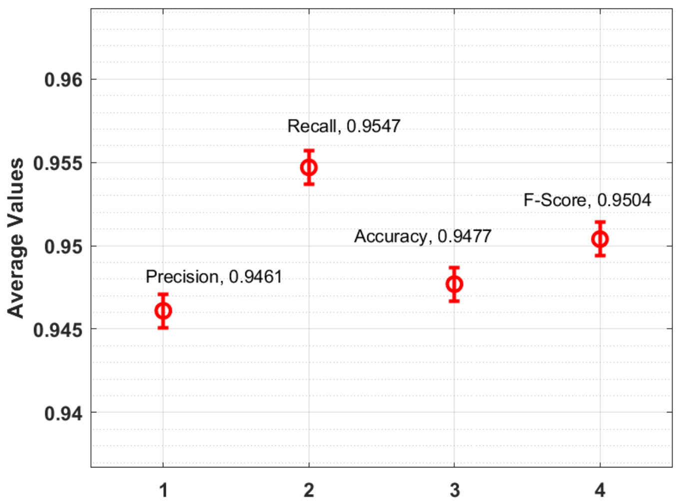

Fig. 5. Average analysis of CSOML-SASC approach

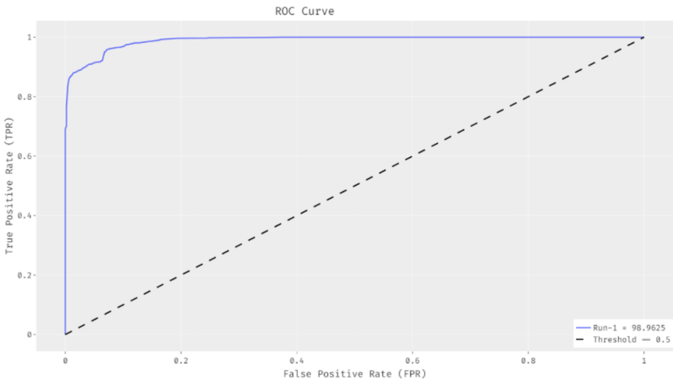

Fig. 6. ROC analysis of CSOML-SASC model under run-1

Fig. 6 demonstrates the ROC analysis of the CSOML-SASC technique under test run-1. The figure depicted that the CSOML-SASC technique has resulted in an increased ROC of 98.9625.

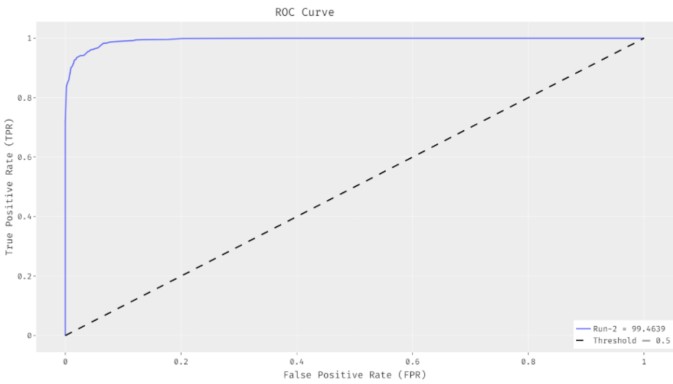

Fig. 7. ROC analysis of CSOML-SASC model under run-2

Fig. 7 showcases the ROC analysis of the CSOML-SASC system in test run-2. The figure demonstrated that the CSOML-SASC algorithm has resulted in a maximum ROC of 99.4639.

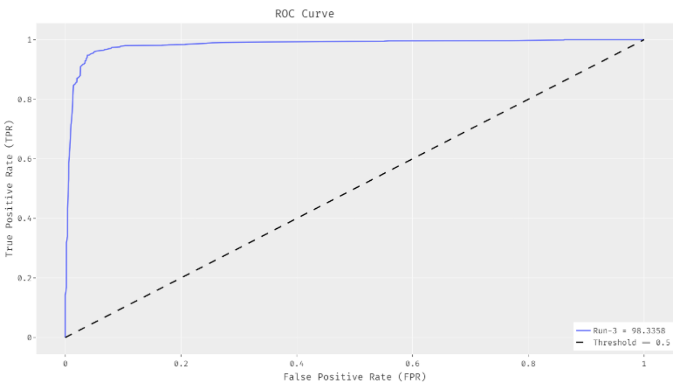

Fig. 8. ROC analysis of CSOML-SASC model under run-3

Fig. 8 illustrates the ROC analysis of the CSOML-SASC manner under test run-3. The figure outperformed that the CSOML-SASC system has resulted in an enhanced ROC of 98.3358.

A comparative results analysis of the CSOML-SASC with existing approaches is provided in Table 2 [22, 23].

Fig. 9 offers the precision and recall analysis of the CSOML-SASC technique on the test dataset. On examining the results in terms of precision, the XLnet, BERT-cased, BERT-uncased, RoBERTa, and DESC techniques have obtained reduced precision values of 71%, 69%, 68%, 78%, and 73% respectively. Besides, the RCNN-RoBERT, LDC, and TCTDF techniques have attained slightly increased outcomes with the moderate precision of 81%, 92%, and 93% respectively. However, the proposed CSOML-SASC technique has resulted in a higher precision of 94.61%.

On investigative the results with respect to recall, the XLnet, BERT-cased, BERT-uncased, RoBERTa, and DESC systems have reached minimal recall values of 72%, 70%, 69%, 79%, and 73% correspondingly. Followed by, the RCNN-RoBERT, LDC, and TCTDF algorithms have achieved somewhat enhanced outcomes with moderate recall of 80%, 73%, and 62% correspondingly. Finally, the presented CSOML-SASC method has resulted in a superior recall of 95.47%.

Table 2 Comparative analysis of CSOML-SASC model with existing approaches

Methods	Precision	Recall	Accuracy	F-Score
XLnet	71.00	72.00	71.00	70.00
BERT-Cased	69.00	70.00	70.00	69.00
BERT-Uncased	68.00	69.00	69.00	68.00
RoBERTa	78.00	79.00	79.00	78.00
DESC	73.00	73.00	74.00	73.00
RCNN-RoBERT	81.00	80.00	82.00	80.00
LDC	92.00	73.00	89.00	82.00
TCTDF approach	93.00	62.00	91.00	74.00
CSOML-SASC	94.61	95.47	94.77	95.04

Fig. 10 provides the accuracy and F-score analysis of the CSOML-SASC method on the test dataset. On exploratory outcomes in terms of accuracy, the XLnet, BERT-cased, BERT-uncased, RoBERTa, and DESC methods have reached minimum accuracy values of 71%, 70%, 69%, 79%, and 74% correspondingly. In addition, the RCNN-RoBERT, LDC, and TCTDF methods have reached somewhat enhanced results with the moderate accuracy of 82%, 89%, and 91% correspondingly. Eventually, the presented CSOML-SASC manner has resulted in a maximum accuracy of 95.47%.

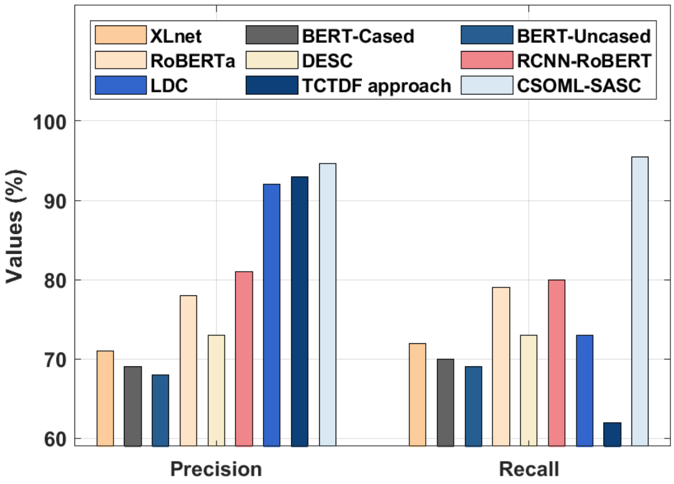

Fig. 9. Precision and recall analysis of CSOML-SASC model

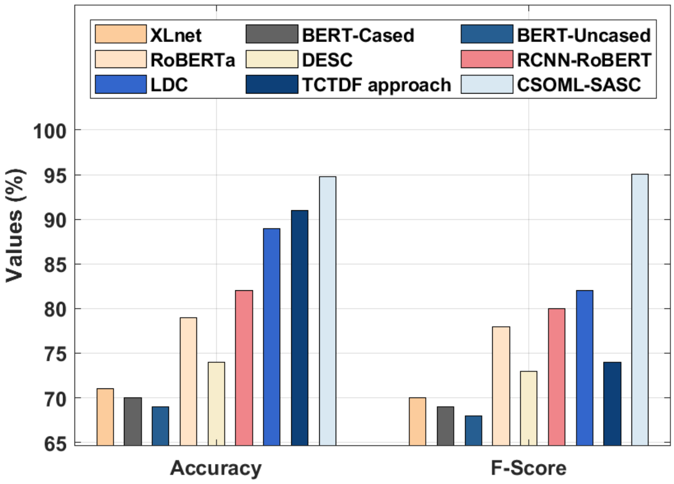

Fig. 10. Accuracy and F-score analysis of CSOML-SASC model

On inspecting the outcomes in terms of F-score, the XLnet, BERT-cased, BERT-uncased, RoBERTa, and DESC manners have reached minimum F-score values of 70%, 69%, 68%, 78%, and 73% correspondingly. In addition, the RCNN-RoBERT, LDC, and TCTDF systems have attained somewhat enhanced results with the moderate F-score of 80%, 82%, and 74% correspondingly. At last, the presented CSOML-SASC methodology has resulted in an increased F-score of 95.04%.

5. CONCLUSIONS

In this study, a novel CSOML-SASC technique is derived for SA on sarcasm detection and classification. The proposed CSOML-SASC technique involves pre-processing, TF-IDF based feature extraction, WKELM based classification, and CSO based parameter optimization. The design of CSO algorithm for tuning the weight and bias values of the WKELM model helps to accomplish improved classification performance. The performance validation of the CSOML-SASC technique takes place on benchmark dataset and the results are inspected under varying aspects. The experimental results pointed out the enhanced outcome of the CSOML-SASC technique over the other recent techniques in terms of different measures. In future, the performance of the CSOML-SASC technique can be boosted by the use of advanced deep learning techniques.

REFERENCES

[1] Lunando E, Purwarianti A. 2013. Indonesian social media sentiment analysis with sarcasm detection. In: 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS). Piscataway: IEEE, 195–198.

[2] Lemmens J, Burtenshaw B, Lotfi E, Markov I, Daelemans W. 2020. Sarcasm detection using an ensemble approach. In: Proceedings of the Second Workshop on Figurative Language Processing. 264–269.

[3] Joshi A, Bhattacharyya P, Carman MJ. 2017. Automatic sarcasm detection: a survey. ACM Computing Surveys 50(5):1–22.

[4] Jena AK, Sinha A, Agarwal R. 2020. C-net: contextual network for sarcasm detection. In: Proceedings of the Second Workshop on Figurative Language Processing. 61–66.

[5] Gupta R, Kumar J, Agrawal H, Kunal. 2020. A statistical approach for sarcasm detection using twitter data. In: 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS). Piscataway: IEEE, 633–638.

[6] Baruah A, Das K, Barbhuiya F, Dey. 2020. Context-aware sarcasm detection using bert. In: Proceedings of the Second Workshop on Figurative Language Processing. 83–87.

[7] Son LH, Kumar A, Sangwan SR, Arora A, Nayyar A, Abdel-Basset M. 2019. Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. IEEE Access 7:23319–23328.

[8] Abdi A, Shamsuddin SM, Hasan S, Piran J. 2019. Deep learning-based sentiment classification of evaluative text based on multi-feature fusion. Information Processing & Management 56(4):1245–1259.

[9] Mandal, P.K. and Mahto, R., 2019. Deep CNN-LSTM with word embeddings for news headline sarcasm detection. In 16th International Conference on Information Technology-New Generations (ITNG 2019) (pp. 495-498). Springer, Cham.

[10] Mohammed, P., Eid, Y., Badawy, M. and Hassan, A., 2019, October. Evaluation of Different Sarcasm Detection Models for Arabic News Headlines. In International Conference on Advanced Intelligent Systems and Informatics (pp. 418-426). Springer, Cham.

[11] Bharti, S.K., Babu, K.S. and Jena, S.K., 2017, December. Harnessing online news for sarcasm detection in hindi tweets. In International Conference on Pattern Recognition and Machine Intelligence (pp. 679-686). Springer, Cham.

[12] Farha, I.A. and Magdy, W., 2021, April. Benchmarking Transformer-based Language Models for Arabic Sentiment and Sarcasm Detection. In Proceedings of

the Sixth Arabic Natural Language Processing Workshop (pp. 21-31).

[13] 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).

[14] Chudi-Iwueze, O. and Afli, H., 2020. Detecting Sarcasm in News Headlines. In CERC (pp. 100-111).

[15] Shrikhande, P., Setty, V. and Sahani, A., 2020, November. Sarcasm detection in newspaper headlines. In 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS) (pp. 483-487). IEEE.

[16] Jamil, R., Ashraf, I., Rustam, F., Saad, E., Mehmood, A. and Choi, G.S., 2021. Detecting sarcasm in multi-domain datasets using convolutional neural networks and long short-term memory network model. PeerJ Computer Science, 7, p.e645.

[17] Zhang W, Yoshida T, Tang X. 2011. A comparative study of TF* IDF, LSI and multi-words for text classification. Expert Systems with Applications 38(3):2758–2765.

[18] Yu, X., Feng, Y., Gao, Y., Jia, Y. and Mei, S., 2021. Dual-Weighted Kernel Extreme Learning Machine for Hyperspectral Imagery Classification. Remote Sensing, 13(3), p.508.

[19] Malik Shehadeh Braik, Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems, Expert Syst. Appl. 174 (2021) 114685.

[20] Umamageswari, A., Bharathiraja, N. and Irene, D.S., 2021. A Novel Fuzzy C-Means based Chameleon Swarm Algorithm for Segmentation and Progressive Neural Architecture Search for Plant Disease Classification. ICT Express.

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

[22] Potamias, R.A., Siolas, G. and Stafylopatis, A.G., 2020. A transformer-based approach to irony and sarcasm detection. Neural Computing and Applications, 32(23), pp.17309-17320.

[23] Bharti, S., Vachha, B., Pradhan, R., Babu, K. and Jena, S., 2016. Sarcastic sentiment detection in tweets streamed in real time: a big data approach. Digital Communications and Networks 2 (3): 108–121.

BIOGRAPHIES

Abel Sridharan has a degree in Computer Science and Post Graduate Program in Data Science and Business Analytics from the University of Texas at Austin (Texas McCombs) and Great Lakes Institute, India. He is currently working as a Senior Digital Engineering Manager building products that directly impact consumers across the globe. He also involves himself in independent research in the areas of Digital Technologies, Data Science, Machine Learning, Artificial Intelligence, Cloud Computing.