IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 6, December 2025, pp. 5027~5037
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i6.pp5027-5037 ❒ 5027

Metaheuristic optimization for sarcasm detection in social media with embedding and padding techniques

Geeta Sahu[1], Manoj Hudnurkar[2]
[1]Department of Information Technology and Data Science, Vidyalankar School of Information Technology, Mumbai, India
[2]Operations and Analytics, Symbiosis Centre for Management and Human Resource Development (SCMHRD), Symbiosis International (Deemed University), Pune, India

Article Info

Article history:

Received Dec 18, 2024
Revised Oct 3, 2025
Accepted Oct 16, 2025

Keywords:

Deep convolution neural network
Embedding
Ensemble classifier
Optimal features
Padding
Sarcasm identification
Weight optimization

ABSTRACT

Sarcasm is a sophisticated mode of expression that allows speakers to express their opinions subtly. Stakeholders provide unstructured messages with extended phrases, making it difficult for computers and people to understand. This research aims to develop a sarcasm detection method to identify words in phrases as sarcastic or non-sarcastic from text, utilizing natural language processing appliances. The first step is pre-processing, when the padding and embedding are performed. Zero padding and end padding are used for the padding. At the same time, different embedding techniques, such as word2vec, Glove, and BERT, are used. Following pre-processing, the features are extracted from the pre-processed data, including "information gain, chi-square, mutual information, and symmetrical uncertainty-based features." Then, a hybrid optimization technique known as clan-updated grey wolf optimization (CU-GWO) is used for optimized features and weight selection. An ensemble technique was applied to extract optimal features. The classifiers in the proposed suggested ensemble technique with deep convolution neural network (DCNN). DCNN offers fine weight tuning and detection results. The performance analysis and its impact on the proposed model for sarcasm detection are classified with good accuracy into sarcastic and non-sarcastic categories. The results are also compared with against those of the GloVe and BERT techniques.

Corresponding Author:

Geeta Sahu
Department of Information Technology and Data Science, Vidyalankar School of Information Technology
Mumbai, India
Email: geeta.sahu@vsit.edu.in

1. INTRODUCTION

Sarcasm examines and classifies an individual's thoughts, sentiments, and emotions into three categories: positive, negative, and neutral. People communicate their sentiments and feelings in various ways. Social networks and microblogging websites frequently employ sarcasm, a sophisticated kind of irony, because they often promote trolling and criticism of other users. Sarcastic evaluation is critical for obtaining meaningful information from amorphous data sources such as reviews and tweets [1]. Sarcasm is a sentimentality category used to communicate good, negative, and neutral moods through words, text, or sentences [2], [3]. Sarcasm is often perceived as a language element that characterizes online content and expresses strongly held opinions and subjectivity [4].

Sentiment analysis, a technique used in advertising and opinion mining, is valuable for determining attitudes. Therefore, sarcasm detection precedes that of the primary NLP devices [5]. Users can exchange information and thoughts on social media platforms [6]. Social media users can submit content, including photographs, videos, and words, regarding any situation to express their feelings [7]. However, negative text comments undermine the excellent sentiment of social media users. Sarcasm is a verbal expression style in which words are used to offend or offend another person by departing from their literal meanings. People often become furious and condemn them [8]. Sarcasm can be identified in written text, gestures, facial expressions, and other forms of communication. Sarcasm is a combination of both positive and negative observations [9]. Sarcasm's interpretation and usage are greatly influenced by the cultural context in which it is employed. This may induce variations in the examination of the entire polarity and alter the polarity of the expression while it is being evaluated. Sentiment analysis is essential for self-examining conduct and assessing sentiments expressed on social media using social listening techniques [10].

Detecting sarcasm in unstructured text or speech, such as instructions, blogs, comments on individuals or events, and product or service evaluations, is known as sarcasm detection. It also enhances human-machine communication efficacy by providing insights into an individual's emotions, psychology, and, occasionally even health. Sentiment types are well defined: no matter whom you ask or what language you use, love is always a positive sentiment, and hatred is always a negative sentiment [11].

Hashtags often accompany a sarcastically worded sentence. For instance, "I love boring food. #not." Saying "I adore bland food" in this context might not be ironic. However, there is sarcasm in "#not," NLP ought to make the data visible using hashtags. Therefore, it is essential to recognize irony in everyday contact and discourse to avoid misunderstandings and enhance sentimental meaning [12]. Sarcasm detection has been achieved using numerous machine learning and artificial intelligence algorithms, such as support vector machine (SVM), random forest (RF), convolution neural network (CNN), and neural network (NN) [13].

They proposed a bidirectional long short-term memory (Bi-LSTM) model to identify sarcasm. The dataset was acquired from Kaggle [14]. News headlines were included in the collection. The dataset was pre-processed by removing special symbols, lowering, stopping word removal, tokenization, and punctuation, followed by word embedding to obtain word vectors with GloVe. The vectors generated using this method were fed into the Bi-LSTM model.

Proposed a novel ensemble method based on text embedding using fuzzy evolutionary logic in the top layer. This method utilizes fuzzy logic to ensemble embedding from the word to vector (Word2vec), global vectors for word representation (GloVe), and bidirectional encoder representations from transformers (BERT) models before deciding on the ultimate categorization [15]. The three social media datasets used to validate the suggested model were the headlines dataset, the "self-annotated reddit corpus" (SARC), and the X (formerly Twitter) app dataset. Accuracy rates of 90.81%, 85.38%, and 86.80% were obtained, respectively [16], [17]. A novel method for implementing a hybrid optimization strategy helps to identify sarcasm [18]. Contextualizing a term demands fluency through articulation [19]. Typically, sentiment characteristics are manually built using a sarcasm-detection algorithm [20].

Deepfake video detectors examine voice tones, lip-sync, facial expressions, and micro expressions. Sarcasm detectors, on the other hand, examine textual content, tone, gestures, and facial expressions, among other things. CU-GWO optimization techniques aid in selecting important characteristics and fine-tuning model parameters. Disambiguation: depending on the context, words like "great," "love," or "fine" might have quite different meanings. capturing syntactic and sequential dependencies: contextual and transformer models are adjusted according on platform-specific information.

The approach employed is dynamic contextual modulation and emotion-embedded vectors to capture the text's emotional content, both the model's contextual adaptation and the hierarchical attention mechanism for text segmentation. The model obtained an F1-score of 0.90 and an accuracy of 89% on the Mustard dataset. Proposed model utilizes the nature inspired swarm-based bio hybrid optimization techniques, mingled-elephant herding & grey wolf optimization (GWO), i.e. Clan updated GWO. The positive Metric-F_measure evaluated over Word2vec, Glove, and BERT are 93%, 91%, and 90%, respectively. This article uses audio data from spontaneous, real-world, monolingual datasets to attempt to detect sarcasm in speech. Irony, exaggeration, subtlety, semantics, and pragmatics are not taken into account. The dataset is small, and there is no baseline comparison.

This work presents sarcasm identification from social media using the innovative CU-GWO-based deep ensemble technique, a hybrid nature inspired swarm based optimization technique, i.e. elephant herding optimization (EHO) + GWO, with pre-processing techniques such as embedding (Word2vec, Glove, and BERT) and padding (zero and end), feature extraction with optimal feature selection, followed by ensemble classifier with RF, SVM, and NN, the output of these classifiers given as input to deep convolution neural network (DCNN), weight optimization is done and DCNN produces the desired result. Comparing the model’s results at various stages with and without feature, weight optimization and with feature, weight optimization. The evaluation of performance for sarcasm type detection from social media using embedding and padding techniques.

2. METHOD

Steps in identifying sarcasm from text are pre-processing, embedding and padding techniques, feature extraction, ideal feature selection, optimized weight selection, and deep learning-based detection methods. In the initial pre-processing stage, tokenization was on the dataset. The keywords are then extracted from each domain. This is followed by embedding, such as Word2vec, Glove, and BERT, known as the word to vector, which is used, followed by zero and end padding. Many features are extracted, such as symmetrical uncertainty-based features, mutual information, chi-square, and information gain. The best characteristics are fed into a combination approach that combines SVM, NN, DCNN, and RF to detect sarcastic statements in the input text. This work suggests a new CU-GWO hybrid model (EHO+GWO) that includes weight adjustment using a DCNN, searching for optimal features. The objective function of the DCNN model is determined with features and weights, as given in (1). Figure 1 depicts the architecture of the proposed sarcasm detection system.

$$Obj = Min\ (Loss) \tag{1}$$

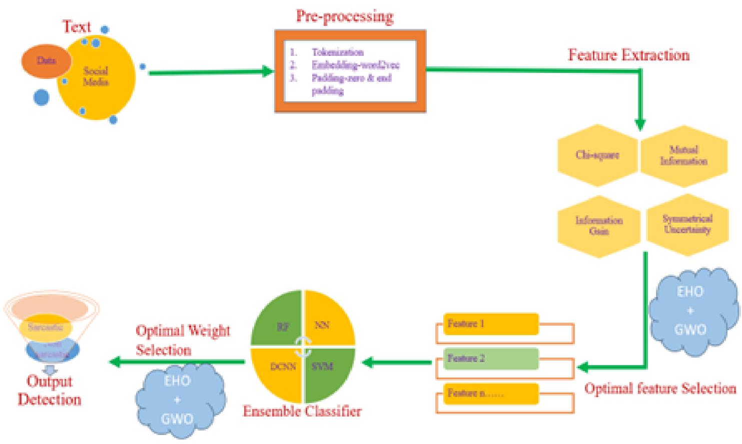

Figure 1. Architecture of the suggested sarcasm detection system

2.1. Pre-processing

Pre-processing steps applied to the data: cleaning text. One major disadvantage of using X data sets is the noise in the data. User mentions (@user), uniform resource locator references (URL), introductory text, and content tags (#), commonly referred to as hashtags, make up X data, or tweets. In this step, X data is pre-processed before feature extraction and classification. Additional basic pre-processing techniques that change the input text to lowercase include tokenization, stop word removal, and parts of speech (POS) tagging. Keyword extraction and stop word removal: in this initial step, each domain's stop words are eliminated before the keywords are recovered. Tokenization: specific words or phrases are denoted by tokens. Tokenization is used to divide a text stream into discrete tokens. Vectorization is the process of representing each sentence or text as a vector, where each element represents a word in the vocabulary and the value of each component indicates where that word appears in the document. This is sometimes referred to as term frequency (TF). E.g., "I cannot believe you did that!" can be expressed in terms of [0, 1, 1, 0, 0, 1, 1].

2.2. Feature extraction

Feature extraction process: since the features are selected and extracted according to how well they fit within the data type, chi-square, information gain, and related features. To identify sarcasm, the subtle linguistic, syntactic, semantic, and pragmatic information presented in a high-dimension space is examined. Chi-square evaluates the degree of independence between a feature and the target variable. By measuring the reduction in entropy, features are exploited, and information gain is quantified. The following extraction methods are utilized.

2.2.1. Chi-square (χ^2):

Chi-square, whose value is determined by dividing the target feature by the remaining feature [21]. Therefore, in (2) expresses determining the best chi-square value/feature (χ^2_f) and choosing the attributes where OF *is the* observed frequency and EF is the expected frequency [22].

$$\chi^2 f = \frac{(OF-EF)^2}{EF} \tag{2}$$

2.2.2. Mutual information

The computation of the shared information between two combinations of random variables, a and b, is called the mutual information feature (MI_f). The MI_f are extracted and shown in (3), where p is the probability.

$$MIf = \sum [\, p\,(a,b) \times log\, 2\, (\frac{p(a,b)}{p(a)\times p(b)})\,] \tag{3}$$

2.2.3. Information gain features

In a and b are random variables, E is entropy, and w is weight [23]. The extracted feature information gain features (IG_f) are calculated using (4) and is given as:

$$IGf = E\,(a) - (w \times E(b)) \tag{4}$$

2.2.4. Features with symmetric uncertainty based

It is expected, and the extraction of how features are extracted is shown in (5), where E denotes the entropy. SU_f denotes the features extracted based on symmetric uncertainty (SU).

$$SUf = 2 \times (\frac{MI\,(a\,,b)}{E\,(a) * E\,(b)}) \tag{5}$$

Finally, the total of all the extracted features, which include the chi-square, mutual information, information gain, and symmetric uncertainty-based features, is represented by the letter F, which can be found in (6).

$$F = [\, \chi^2 f + MIf + IGf + SUf\,] \tag{6}$$

Interaction with the optimization technique: for the feature optimization dataset, the CU-GWO model is utilised. It goes through a preprocessing phase first. Following the extraction of all the components, certain traits obstruct the training. What might be referred to as the "curse of dimensionality" affects the model's performance. Balance between exploration and exploitation: CU-GWO improves the equilibrium between exploitation (identifying the best solution with the least loss and maximizing model performance) and exploration (finding several solutions). It accelerates the convergence rate.

2.3. Optimal feature selection and deep ensemble classifier technique for sarcasm recognition

The training procedure became overburdened with pertinent and unnecessary data for all derived features. The term 'curse of dimensionality' has becomes an issue. This study paper provides a new meta-heuristic-based bio-inspired hybrid optimization approach (CU-GWO) to determine the best characteristics. Figure 2 illustrates the best and optimal feature selection.

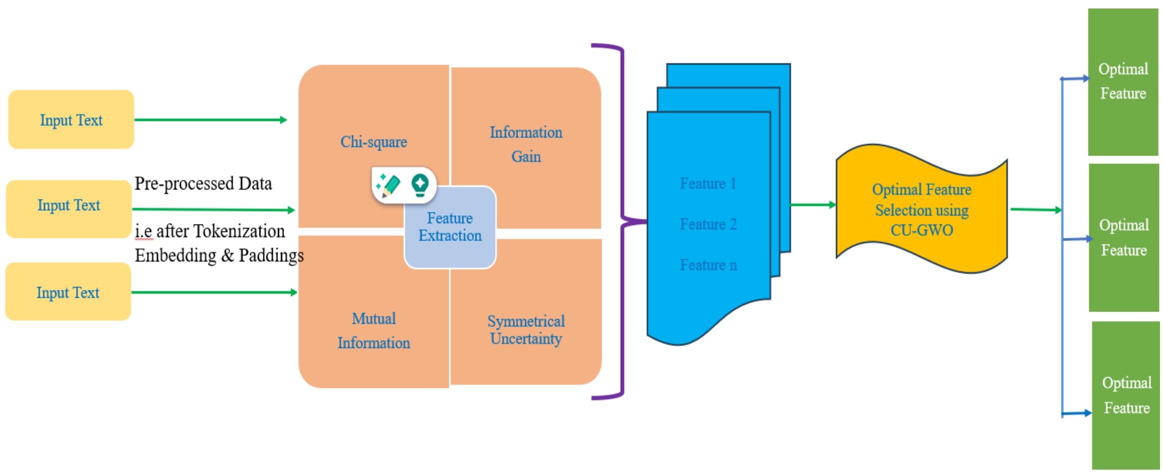

Figure 2. Optimal feature selection

2.4. Ensemble classifiers

The developed ensemble technique uses various classifiers, including DCNN, NN, SVM, and RF. These classifiers were assigned given the best feature selection. The classifier result is then sent to the CNN. The final detection results were assessed [24]. Figure 3 illustrates the ensemble classifier and optimised DCNN classifier. RF: a classification framework uses several weak classifiers to create a robust classifier. The RF model ensures random v, and CART DT uses every feature. Compared with DT, the RF model offers better classification accuracy. SVM is a nonlinear mapping that increases the magnitude of training data from actual training data. It looks for this linear optimal hyperplane of this extra dimension. In (7) expresses the assessment of the hyperplane in two separable class linear problems, where NV is the average vector and hp is the distance between the origins and hyperplanes. SVM denotes the output of the SVM.

$$Svm = NV + hp = 0 \tag{7}$$

Neural network: input for this is the ideal feature F, defined by (8), where F_{fc} denotes the overall feature count.

$$F = [Fa + Fb + Fc + \cdots. + Ffc] \tag{8}$$

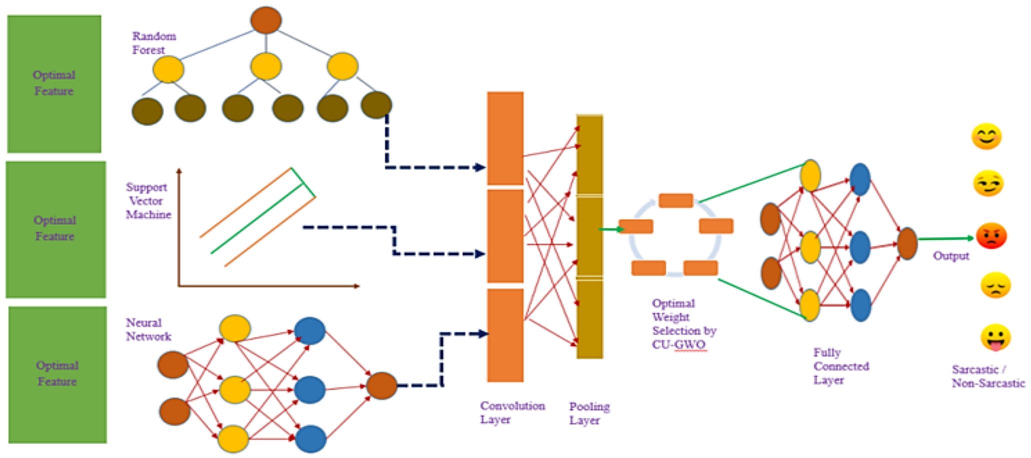

Figure 3. Ensemble classifier and optimised DCNN classifier

2.5. Enhanced deep convolution neural network classification

The feature assessment at that position (x, y) in the corresponding feature map and r[th] layer is expressed by (9), where $P.L^{r}_{x,y}E$ represents the pooling layer function with a fixed input at the layer position (x, y).

$$CF^{\,r}_{x,y,z} = W_z^{\,r\,T} * PL^{\,r}_{x,y} + D_z^r \tag{9}$$

Integration of CU-GWO with DCNN: to reflect the detection of sarcasm in the output, features-lexical cues, sentiment, and contextual information need to be appropriately weighted. Dimensionality reduction is done where weight selection helps lessen the impact of redundant, unnecessary information that could introduce noise into the detection process. Exploration and exploitation: CU-GWO improves the equilibrium between exploitation (identifying the best solution with the least loss and maximizing model performance) and exploration (identifying several solutions). It accelerates the convergence rate and clear of overfitting problems.

2.6. Proposed CU-GWO algorithm for the optimization process

The most critical wolves in the hunting process are represented by alpha (α), beta (β), gamma (¥), and delta (§). Leader α in this group can decide about the hunting process, while wolves β in the second and third positions ¥ support the leader in making decisions [25]. The proposed CU-GWO model is described in (10).

$$CU^{\,it+1}_{ij} = CU^{it}_{ij} + Ø\,(\,ma^{\,it}_{i} - CU^{it}_{ij}\,) + δ * (\,cc^{\,it}_{i} - CU^{it}_{ij}) + ψ * \text{rand} \tag{10}$$

CU-GWO is a better approach compared to existing methods. It is a model with five stages. Two metaheuristic optimization techniques: EHO+GWO. Features of the following types are employed: pragmatic, lexical, syntactic, semantic, and context-based. Selected features are used to get rid of the “curse of dimensionality”. It manages data with non-linear relationships and high dimensions and eliminates noisy and ambiguous data. RF, SVM, NN, and DCNN ensemble classifiers are used. DCNN is used for weight fine-tuning. Optimal weights and fine-tuning are computed.

Exploration and exploitation: CU-GWO improves the equilibrium between exploitation (identifying the best solution with the least loss and maximizing model performance) and exploration (identifying several solutions). Word2vec has the highest sensitivity (0.95) and zero padding. It accelerates the convergence rate & can investigate various features and optimization strategies to enhance the model's performance. The accuracy of the balanced dataset-0.92, the sensitivity was 0.95, and the F-measure was 0.93.

How CU-GWO works: two datasets are included in this model: one was balanced, while the other was unbalanced. The proposed method for identifying sarcasm in the text dataset consists of five steps: preprocessing, appropriate feature extraction, ideal feature selection, and deep learning-based detection. The dataset first undergoes a preprocessing step that includes tokenization and stop-word removal. The best-selected features are fed into a combination approach that consists of RF, SVM, NN, and DCNN, which then generates the intended outcome.

3. RESULTS AND DISCUSSION

Python was used to simulate the method [26]. MUStARD consists of audio-visual statements labeled with sarcasm identification. The input parameters are: alpha =0.025, embedded_sentence size-26, model=word2vec. Word2Vec (vector_size =100, vocabulary =2965). Balanced dataset dimension is (690,3). Imbalanced dataset - file 1 sarcastic dimension (200,3), file 2 non-sarcastic dimension (300,3). Imbalanced - file 1, file 2=300+200=500=(500,126). Training set (350,126), testing set (207,126). [60%, 40%]. Tables 1 and 2 describe the parameters & give the statistical parameters on a balanced dataset for training and testing.

Table 1. Experimental parameters

Metrics	Values
Features	Headlines (text)
Records	26,691
Sarcastic records	13,568
Without sarcasm	13,123
Epoch	25
Size of population	5
EHO	n_clans =5, alpha =0.5, beta =0.5
WOA	b =1 and p =0.5

Table 2. Statistical parameters on a balanced dataset for training and testing

Balanced dataset	Word2vec (70%, 30%)	Glove	BERT
690	train_data_m (483, 126) test_data_m (207, 126)	train_data_m (483, 99) test_data_m (207, 99)	train_data_m (483, 71) test_data_m (207, 71)

3.1. Operational procedure

The purpose is to use embedding and padding techniques in the proposed CU-GWO algorithm with a DCNN to investigate the recently introduced sarcasm detection model. The recently added steps are as follows:

i) Pre-processing, involves finishing padding and embedding. Word2vec embedding is used for embedding, and zero padding is utilized for padding. After pre-processing, "information gain, chi-square, mutual information, and symmetrical uncertainty-based features" extract the features from the data.
ii) Subsequently, a hybrid optimization method known as clan-updated grey wolf optimization (CU-GWO) is used to choose the best features. The proposed suggested ensemble technique includes the NN, SVM, RF, and DCNN classifiers.
iii) Evaluated the impacts and performance of various embedding techniques, such as Word2vec, GloVe, and BERT in the balanced and unbalanced dataset.

3.2. Tokenization

The dataset first undergoes a preprocessing step that includes tokenization and stop-word removal. The raw data that was gathered was pre-processed using padding and tokenization. Keywords are then pulled from each domain. Once tokenization is achieved, embedding such as Word2vec, known as work to vector, is used, followed by zero and end padding, and padding techniques utilized.

3.3. Feature extraction

In supervised learning, feature extraction comes after the pre-processing phase. A set of features is created from the data. Among the traits are psychological, linguistic, emotional, pragmatic, and exaggerated traits. After pre-processing, the tokenized words undergo feature extraction using the Chi-square, mutual information, information gain, and symmetrical uncertainty techniques.

3.4. Evaluation of performance utilizing Word2vec embedding and zero, end padding with CU-GWO

Techniques for padding and embedding are essential-embedding is the process of representing words or tokens in vector space as numbers. To transform text input into a numerical format suitable for machine learning, we use Word2vec, Glove, and BERT. By allowing pre-trained embedding to different tasks, transfer learning is used to improve generalization and save computation. Dense vectors reduce the high-dimensional character text data, optimizing, increasing the efficiency of complex operations. Padding techniques help in truncation loss, efficient batch processing & memory management. The accuracies achieved in Word2vec, Glove, and BERT are 92%, 90%, and 89%, respectively. The positive Metric-F_measure evaluated over Word2vec, Glove, and BERT is 93%, 91%, and 90%, respectively. The negative metric-FPR values for Word2vec, Glove, and BERT are 0.11, 0.095, and 0.14 for zero padding. Figure 4 the performance, and Figure 5 presents character samples, with Figure 5(a) showing the characters prior to preprocessing, and Figures 5(b) to 5(k) showing the characters following preprocessing.

3.5. Evaluation of performance on the optimal features and weight selection

The effectiveness of CU-GWO (EHO+GWO) works with various training data, utilizing imbalanced and balanced datasets for sarcasm detection based on the best features, and weight selection is performed [27]. The model performs better in sarcasm detection, achieved an accuracy rate of 91%, 91.2%, and 92.75%, respectively. Table 3 depicts the CU-GWO comparison with and without feature and weight optimizations.

Table 3. CU-GWO comparison with and without feature and weight optimizations

Measures	CU-GWO without features selection	CU-GWO without weight optimization	CU-GWO without feature selection or weight optimization	CU-GWO with weight optimization and feature selection
Accuracy	0.853	0.879	0.826087	0.927536
Matthew's correlation coefficient (MCC)	0.804	0.828	0.649375	0.850965
False discovery rate (FDR)	0.140	0.114	0.151786	0.073171
Sensitivity	0.854	0.880	0.833333	0.950
False positive rate (FPR)	0.148	0.122	0.182796	0.103448
Negative predictive value (NPV)	0.846	0.871	0.800	0.928571
Specificity	0.852	0.878	0.817204	0.896552
The Rand index	0.878	0.904	0.908893	0.963087
Precision	0.860	0.886	0.848214	0.926829
False negative rate (FNR)	0.146	0.120	0.166667	0.050
F-measure	0.857	0.883	0.840708	0.938272

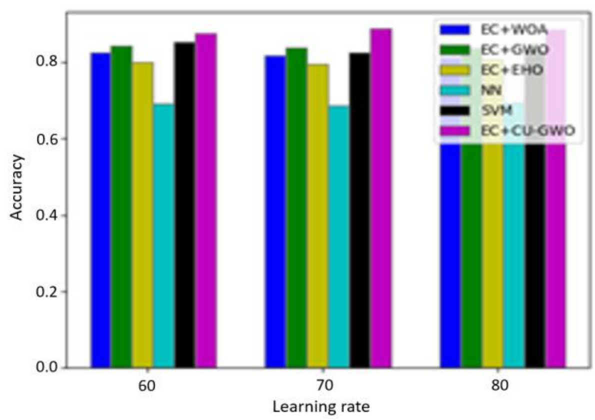

Figure 4. The graph indicates the proposed model is tested and performance is achieved in terms of accuracy

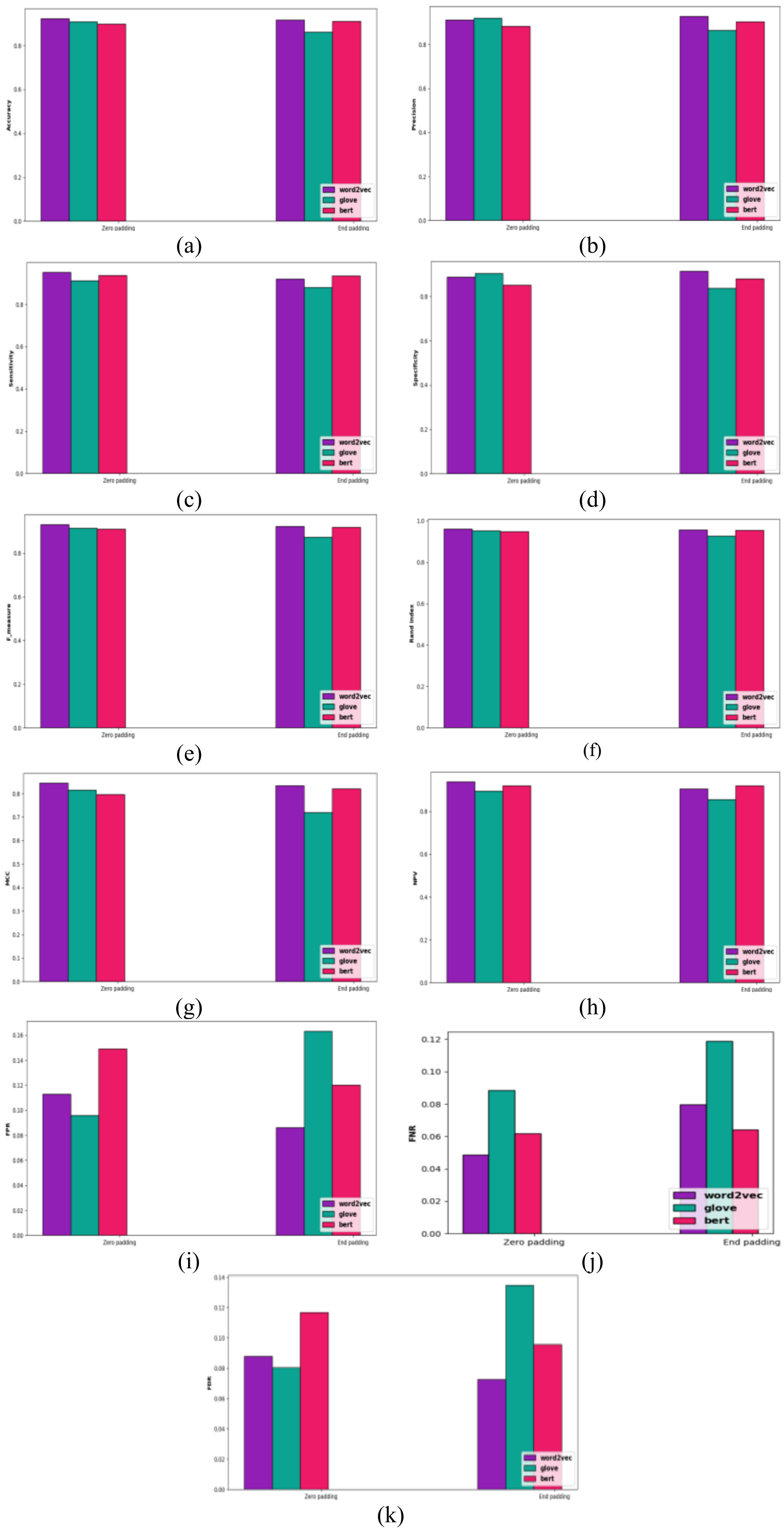

Figure 5. Character samples were taken from (a) accuracy, (b) precision, (c) sensitivity, (d) specificity, (e) f_measure, (f) rand_index, (g) MCC, (h) NPV, (i) FPR, (j) FNR, and (k) FDR

3.6. Confusion matrix

A table used to assess a classification model's performance is called a confusion matrix. To describe the categorization findings, it contrasts the actual class labels with the predicted class labels. A confusion matrix is used to assess how well a classification model performs. The actual and anticipated values in the matrix are compared by the model. Table 4 shows the evaluation of confusion matrix.

Table 4. Confusion matrix values

Model	TN	FP	FN	TP
Word2Vec – Zero padding	87	7	10	103
GloVe – Zero padding	81	13	14	99
BERT – Zero padding	87	6	12	102
Word2Vec – End padding (set 1)	82	11	5	109
Word2Vec – End padding (set 2)	82	11	9	105
Word2Vec – End padding (set 3)	82	11	10	104

3.7. Performance evaluation with Word2vec embedding and zero, end padding

A balanced dataset provides more accurate results of 0.9221 compared to an imbalanced dataset 0.8985. FDR in imbalance dataset is 0.12903 and balanced dataset is 0.08800. Table 5 shows the evaluation. Benefits of CU-GWO versus alternative optimization methods: exploration and exploitation ensure that the optimizers explore a larger solution space for feature weights and model parameter adjustment, preventing premature convergence. Multimodal feature optimization technique is excellent for conducting effective searches in intricate multimodal feature spaces. By optimizing both feature selection and decision fusion procedures, it offers improved fitness evaluation. Algorithms such as differential evolution and particle swarm optimization become trapped in local optima and are unable to maintain equilibrium across modalities.

Table 5. Evaluation of performance utilizing Word2vec embedding and zero, end padding

Method	Accuracy	Sensitivity	Specificity	Precision	FNR	F measure	Rand index	MCC	FPR	NPV	FDR
Imbalance data	0.898	0.955	0.829	0.870	0.044	0.911	0.947	0.798	0.170	0.939	0.129
Balanced data	0.922	0.951	0.887	0.911	0.048	0.930	0.960	0.844	0.112	0.938	0.088

4. CONCLUSION

A five-phase model for sarcasm detection is proposed. Pre-processing is to the text input, which includes tokenization, embedding, and padding. Embedding includes Word2vec, Glove, and BERT, followed by padding. CU-GWO was utilized. NN, SVM, RF, and DCNN were classifiers used in ensemble approaches. The DCNN results indicated an accuracy of 0.922. A balanced dataset yields more accurate results than unbalanced. Limitations: the annotation procedure was more difficult because annotators had to rely on just the public dataset. Misunderstandings could result from machine learning algorithms' inability to grasp context's subtleties. A model trained in one cultural setting might not perform well in another. The efficiency of the model can be improved by including visual commonsense information from image captions and backdrop descriptions that is not included. Due to the wide and diverse nature of the input, the current model has not taken into account answers, review ranking systems, or dialog. Future research can look into developing sarcasm recognition models that are region-specific and domain-specific (politics or entertainment). It can be expanded to multimedia computing applications as real-time voice interpretation, deepfake video detection, emotion recognition, and the internet of multimedia things. Uses of large language models (LLMs) for sarcasm detection will be investigated in future studies. Images, audio, video, emoticons, and memes can all be used to identify sarcasm. Future studies can also look into the financial markets, depression, defamation, government issues, cyberstalking, and cyberbullying.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Geeta Sahu	✓	✓		✓	✓	✓			✓	✓	✓			
Manoj Hudnurkar	✓	✓				✓			✓	✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available in GitHub at https://github.com/soujanyaporia/MUStARD/blob/master/data/sarcasm_data.json.

REFERENCES

[1] A. Kalaivani and D. Thenmozhi, "Sarcasm identification and detection in conversion context using BERT," in *Proceedings of the Second Workshop on Figurative Language Processing*, 2020, pp. 72–76, doi: 10.18653/v1/P17.

[2] Y. Du, T. Li, M. S. Pathan, H. K. Teklehaimanot, and Z. Yang, "An effective sarcasm detection approach based on sentimental context and individual expression habits," *Cognitive Computation*, vol. 14, pp. 78–90, 2022, doi: 10.1007/s12559-021-09832-x.

[3] R. Justo, T. Corcoran, S. M. Lukin, M. Walker, and M. I. Torres, "Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web," *Knowledge-Based Systems*, vol. 69, pp. 124–133, 2014, doi: 10.1016/j.knosys.2014.05.021.

[4] A.-M. Iddrisu, S. Mensah, F. Boafo, G. R. Yeluripati, and P. Kudjo, "A sentiment analysis framework to classify instances of sarcastic sentiments within the aviation sector," *International Journal of Information Management Data Insights*, vol. 3, no. 2, Nov. 2023, doi: 10.1016/j.jjimei.2023.100180.

[5] A. Kumar, S. Dikshit, and V. H. C. Albuquerque, "Explainable artificial intelligence for sarcasm detection in dialogues," *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, Jul. 2021, doi: 10.1155/2021/2939334.

[6] R. Kumar and J. Kaur, "Random forest-based sarcastic tweet classification using multiple feature collection," in *Multimedia Big Data Computing for IoT Applications, Intelligent Systems*, vol. 163, Singapore: Springer, 2020, doi: 10.1007/978-981-13-8759-3_5.

[7] V. Govindan and V. Balakrishnan, "A machine learning approach in analysing the effect of hyperboles using negative sentiment tweets for sarcasm detection," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 5110–5120, 2022, doi: 10.1016/j.jksuci.2022.01.008.

[8] K. Nimala, R. Jebakumar, and M. Saravanan, "Sentiment topic sarcasm mixture model to distinguish sarcasm prevalent topics based on the sentiment bearing words in the tweets," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 6801–6810, Jul. 2021, doi: 10.1007/s12652-020-02315-1.

[9] Y. Y. Tan, C. Chow, J. Kanesan, J. H. Chuah, and Y. Lim, "Sentiment analysis and sarcasm detection using deep multi task learning," *Wireless Personal Communications*, vol. 129, pp. 2213–2237, 2023, doi: 10.1007/s11277-023-10235-4.

[10] C. I. Eke, A. A. Norman, and L. Shuib, "Context-based feature technique for sarcasm identification in benchmark datasets using deep learning and BERT model," *IEEE Access*, vol. 9, pp. 48501–48518, 2021, doi: 10.1109/ACCESS.2021.3068323.

[11] R. A. Bagate and R. Suguna, "Sarcasm detection of tweets without #sarcasm: data science approach," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 993–1001, Aug. 2021, doi: 10.11591/ijeecs.v23.i2.pp993-1001.

[12] A. Agarwal, A. K. Jha, A. Jaiswal, and V. Kumar, "Irony detection using transformers," in *2020 International Conference on Computing and Data Science (CDS)*, Stanford, CA, USA: IEEE, 2020, pp. 165-168, doi: 10.1109/CDS49703.2020.00040.

[13] K. Sundararajan and A. Palanisamy, "Multi-rule based ensemble feature selection model for sarcasm type detection in Twitter," *Computational Intelligence and Neuroscience*, vol. 2020, no. 1, Jan. 2020, doi: 10.1155/2020/2860479.

[14] R. Misra and P. Arora, "Sarcasm detection using news headlines dataset," *AI Open*, vol. 4, pp. 13–18, 2023, doi: 10.1016/j.aiopen.2023.01.001.

[15] Y. Zhang, Y. Liu, Q. Li, and P. Tiwari, "CFN: A complex-valued fuzzy network for sarcasm detection in conversations," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 12, pp. 3696–3710, Apr. 2021, doi: 10.1109/TFUZZ.2021.3072492.

[16] D. K. Sharma, B. Singh, S. Agarwal, N. Pachauri, A. A. Alhussan, and H. A. Abdallah, "Sarcasm detection over social media platforms using hybrid ensemble model with fuzzy logic," *Electronics*, vol. 12, no. 4, 2023, doi: 10.3390/electronics12040937.

[17] I. A. Farha and W. Magdy, "From Arabic sentiment analysis to sarcasm detection: the ArSarcasm dataset," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools*, Marseille, France: European Language Resource Association, 2020, pp. 32–39.

[18] I. Touahri and A. Mazroui, "Enhancement of a multi-dialectal sentiment analysis system by the detection of the implied sarcastic features," *Knowledge-Based Systems*, vol. 227, 2021, doi: 10.1016/j.knosys.2021.107232.

[19] A. Onan and M. A. Tocoglu, "A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification," *IEEE Access*, vol. 9, pp. 7701–7722, 2021, doi: 10.1109/ACCESS.2021.3049734.

[20] B. N. Hiremath and M. M. Patil, "Sarcasm detection using cognitive features of visual data by learning model," *Expert Systems With Applications*, vol. 184, 2021, doi: 10.1016/j.eswa.2021.115476.
[21] B. Liu, "Sentiment analysis: mining opinions, sentiments, and emotions," *Computational Linguistics*, vol. 42, no. 3, pp. 595–598, 2016, doi: 10.1162/COLI_r_00259.
[22] G. A. Sahu and M. Hudnurkar, "Metaheuristic-assisted deep ensemble technique for identifying sarcasm from social media data," *International Journal of Wireless and Mobile Computing*, vol. 26, no. 1, pp. 25–38, Feb. 2024, doi: 10.1504/IJWMC.2024.136558.
[23] G. A. Sahu and M. Hudnurkar, "Sarcasm detection: a review, synthesis and future research agenda," *International Journal of Image and Graphics*, vol. 23, no. 6, 2023, doi: 10.1142/S0219467823500614.
[24] G. A. Sahu and D. M. Hudnurkar, "Performance analysis of clan updated grey wolf optimization (CU-GWO) based deep ensemble technique for identifying sarcasm from social media data," in *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 2024, pp. 1-10, doi: 10.1109/ICSES60034.2023.10465426.
[25] A. F. Alrasheedi, "Different type of pay later facility for green product with selling price dependent demand using grey wolf optimizer," *Heliyon*, vol. 10, no. 11, 2024, doi: 10.1016/j.heliyon.2024.e32398.
[26] S. Poriya, "MUStARD data," *GitHub.com,* 2020. [Online]. Available: https://github.com/soujanyaporia/MUStARD/blob/master/data/sarcasm_data.json
[27] A. Banerjee, M. Bhattacharjee, K. Ghosh, and S. Chatterjee, "Synthetic minority oversampling in addressing imbalanced sarcasm detection in social media," *Multimedia Tools and Applications*, vol. 79, pp. 35995–36031, 2020, doi: 10.1007/s11042-020-09138-4.

BIOGRAPHIES OF AUTHORS

Dr. Geeta Sahu is awarded a doctoral degree under the Faculty of Computer Studies (FoCS) at Symbiosis Center for Research and Innovation (SCRI), Symbiosis International (Deemed University), Pune. She specializes in deep learning, machine learning, and artificial intelligence. She holds the position of assistant professor, with 16 years of work experience at Vidyalankar School of Information Technology (VSIT), Mumbai, India. She can be contacted at email: geeta.sahu@vsit.edu.in or geetasahu23@gmail.com.

Dr. Manoj Hudnurkar is a Professor of Operations and Analytics at Symbiosis Centre for Management and Human Resource Development (SCMHRD), Symbiosis International (Deemed University), Pune, India. He has conducted numerous research works on supplier performance management, collaborative supply chain management, and supplier classification, which have appeared in reputable international publications. He has also worked on many projects under supply chain management and operations. He can be contacted at email: manoj_hudnurkar@scmhrd.edu.