LEX LOCALIS-JOURNAL OF LOCAL SELF-GOVERNMENT
ISSN:1581-5374 E-ISSN:1855-363X
VOL. 24, NO. 3 (2026)

# Fasttext Deep Recurrent Network And Optimized Densenet Based Aggregated Max Fusion For Multimodal Sarcasm Detection In Local Self-Government Social Media Platforms

**Mr. K. Sathyaseelan[*1] , Dr. N. Yuvaraj [2]**

[1]* Research Scholar, Department of Computer Science and Engineering, Erode Sengunthar Engineering College (Autonomous),Thudupathi, Perundurai, Erode-638057,Tamil Nadu, India.
[2]Professor, Department of Computer Science and Engineering, Erode Sengunthar Engineering College (Autonomous),Thudupathi, Perundurai, Erode-638057, Tamil Nadu, India
Corresponding Author: [1]* sathyaseelan086@gmail.com

Received: 16/02/2026; Accepted: 29/06/2026

**Abstract-**Multimodal sarcasm detection is a complex but promising area of research and application, leveraging diverse data types to improve understanding and interpretation of sarcastic expressions. Traditional sarcasm detection often relies solely on textual analysis, which can miss the nuances conveyed through tone of voice or facial expressions. Deep learning models, particularly those leveraging architectures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs), can effectively process and integrate these multiple data modalities. By analyzing patterns across different inputs, these models can improve the accuracy of sarcasm detection, capturing the subtleties that indicate sarcasm, such as intonation or context-specific gestures. In this paper, Pearson Correlated FastText and Bray Curtis Deep Recurrent Neural Network (PCFT-BDRNN) model is proposed for detecting sarcasm using text data on social media. Initially, number of text data from the given dataset is collected as input. After that, text preprocessing using Gensim tokenizer, stop-word filtering, word stemming and Lemmatization is carried out. Subsequently, FastText word embedding is performed to determine the semantic information from the preprocessed results. Later, classification of text is achieved by designing Bray Curtis Deep Recurrent Perceptive Neural Learning with better accuracy. Optimized Canonical DenseNet (OP-DenseNet) Model is proposed for sarcasm detection using images on social media. The designed OP-DenseNet model includes multiple layers such as input layer, convolutional layers, global average pooling layer, fully connected layer and output layer. The input layer gets number of images collected from the given dataset. The multiple convolutional layers perform texture, color, intensity, standard deviation and edge feature extraction. Followed by this, spatial dimensions of the images are minimized using global average pooling layer. In addition, the classification of images is made in fully connected layer using Canonical variate analysis. ReLU activation function provides classification results at the output layer for sarcasm detection. Lastly, multimodal fusion is developed using aggregated max fusion model for improving the sarcasm detection where it combines both the results of text and image classifiers. Experimental evaluation is carried out using Multi-modal sarcasm detection Dataset using different performance metrics. The results show that the proposed models significantly improve the accuracy, precision, recall with minimum time than the state-of-the-art methods.

**Keywords:** Multimodal sarcasm detection, multimodal fusion, Deep Recurrent Neural Network, word embedding, DenseNet.

## 1. Introduction

The significance of multi-modal sarcasm detection lies in its potential to improve natural language understanding, enhance user experience, and reduce miscommunication in human-computer interactions. The application of multi-modal approaches is especially relevant in social media, video content, and

conversational Artificial Intelligent (AI), where sarcasm is prevalent and can lead to misunderstandings if not properly identified. As research advances, these models hold the potential to enhance user interactions in virtual environments, improving communication clarity and emotional understanding in AI systems.

A new multi-modal sarcasm detection called cue learning approaches was designed in [1] to handle the problems such as data scarcity. Though the model improves the reliability, the accuracy of sarcasm detection was not enhanced. The Knowledge Fusion Network (KnowleNet) was introduced in [2] for multimodal sarcasm detection via classifying the positive and negative samples. But, the time involved in the multimodal analysis was not reduced.

An improved fine-tuned language approach was designed in [3] for sarcasm identification through integrating the static and contextualized representations. Word2Vec word embeddings and Bidirectional Encoder Representations from Transformers (BERT) models were developed to train the Arabic resources. However, accurate detection of sarcasm was not improved. Hybrid Neural Network with attention mechanism was presented in [4] that consistently learn sarcastic cues from the text. But, the time for sarcasm detection was not focused.

Sarcasm detection in online comments from social media were developed in [5] through the machine and deep learning models. Sarcasm identification in Headline News using machine and deep learning models were introduced in [6]. Sarcastic and non-sarcastic headlines are determined using developed models. However, the precision and recall was not improved.

Hybrid Classifier-Opposition Learning based Aquila Optimization (HC-OLAO) model was presented in [7] for multimodel sarcasm detection. It performs pre-processing, feature extraction, feature level fusion, and classification. Though accuracy was improved, the better classification was not achieved. Sarcasm detection was performed by pre-trained transformer and CNN to capture context features in [8]. But, the robust classification results were not achieved.

A lightweight depth attention scheme through a self-regulated ConvNet was introduced in [9] for multi-modal sarcasm detection via uncovering visual, acoustic and glossary features. But, the precision was not improved. Convolution and Attention with Bi Directional Gated Recurrent Unit was presented in [10] for Sarcasm Detection. But, the error rate was not reduced.

Local Self-Government institutions increasingly rely on digital platforms such as Facebook, X (Twitter), Instagram, grievance portals, and community discussion forums to communicate with citizens and gather public opinion. These online interactions provide valuable insights into citizen satisfaction, service quality, infrastructure issues, and administrative performance. However, citizens frequently express dissatisfaction through sarcastic comments, making automatic interpretation difficult. Misclassification of sarcastic feedback may lead to incorrect policy decisions and inaccurate public sentiment analysis. Therefore, intelligent multimodal sarcasm detection can assist Local Self-Government institutions in understanding genuine citizen opinions and improving evidence-based governance.

### 1.2 Role in Local Self-Government

The rapid digital transformation of Local Self-Government (LSG) institutions has fundamentally changed the way local authorities communicate with citizens and deliver public services. Municipal corporations, panchayats, urban local bodies, and other decentralized government institutions increasingly rely on digital platforms such as Facebook, X (formerly Twitter), Instagram, mobile applications, and online grievance portals to engage with communities, disseminate public information, and receive feedback regarding civic administration. These platforms generate a vast amount of multimodal data comprising textual comments, images, videos, and other user-generated content that reflects citizens' perceptions of municipal governance and service delivery. However, public responses frequently contain sarcasm, irony, and implicit criticism, making it difficult for conventional sentiment analysis techniques to accurately interpret the true intention behind citizen opinions. Consequently, the ability to detect sarcastic expressions has become increasingly important for developing intelligent digital governance systems that support informed administrative decision-making.

The proposed multimodal sarcasm detection framework offers significant practical value for Local Self-Government by enabling accurate interpretation of citizen feedback across multiple communication channels. The framework can assist local administrators in monitoring public opinion, identifying hidden dissatisfaction, and prioritizing grievances related to essential municipal services such as waste management, drinking water supply, road maintenance, sanitation, street lighting, public transportation, property taxation, and other civic amenities. Furthermore, during emergencies such as floods, pandemics, or natural disasters, local governments depend heavily on digital communication to disseminate critical information and understand public responses. Distinguishing genuine concerns from sarcastic or misleading comments enables authorities to improve crisis communication, allocate resources more effectively, and respond promptly to community needs.

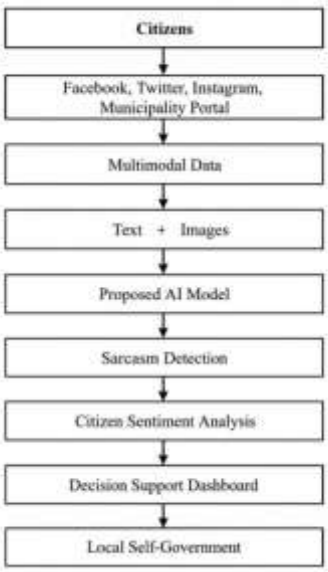


**Figure 1 Governance-Oriented Workflow for Local Self-Government**

By integrating multimodal artificial intelligence into digital governance platforms, the proposed framework supports smart governance, strengthens citizen engagement, enhances transparency and accountability, facilitates evidence-based policy formulation, and promotes responsive, citizen-centric local administration. Therefore, intelligent sarcasm detection can serve as a valuable decision-support tool for Local Self-Government institutions seeking to improve public service quality, strengthen community trust, and foster sustainable and inclusive local governance in an increasingly digital society. Figure 1 illustrates the overall governance-oriented workflow of the proposed multimodal sarcasm detection framework for Local Self-Government. The framework demonstrates how multimodal citizen feedback collected through digital communication platforms is transformed into actionable governance insights that support evidence-based decision-making and improved municipal service delivery.

As illustrated in Figure 1, citizen feedback collected through various digital communication platforms, including Facebook, X (Twitter), Instagram, municipality grievance portals, and mobile applications, is processed as multimodal data consisting of textual comments and images. The proposed artificial intelligence framework performs semantic feature extraction, multimodal sarcasm detection, and sentiment interpretation to identify hidden dissatisfaction expressed by citizens. The analytical outcomes are presented through a decision-support dashboard that enables Local Self-Government authorities to monitor public opinion, prioritize grievances, evaluate municipal service performance, and support transparent, responsive, and evidence-based governance.

### 1.1 Contribution

In order to resolve the issues identified from the existing literature**,** a novel PCFT-BDRNN, OP-DenseNet and Aggregated max fusion model is introduced with the following major contributions.

- A novel model referred as PCFT-BDRNN is proposed for accurate sarcasm detection with different processes such as Gensim tokenizer, stop-word filtering, word stemming and Lemmatization. FastText word embedding is employed to extract the semantic information from text data. Also, Bray Curtis Deep Recurrent Perceptive Neural Learning is designed for sarcasm detection with better precision.
- To enhance the accuracy of sarcasm detection using images, OP-DenseNet model is proposed. Multiple features such as texture, color, intensity, standard deviation and edge features are extracted to minimize the time involved in the sarcasm detection. Canonical variate analysis and ReLU activation function is used to classify the images for sarcasm detection.
- To improve the performance of multimodal sarcasm detection, Aggregated max fusion model is designed to integrate the results from both the text and image classifiers.
- Finally, comprehensive experiments are conducted to estimate the quantitative analysis of the proposed models along with conventional deep learning methods based on various performance metrics.

The rest of this paper is organized into different sections as follows. Section 2 describes related works for multimodal sarcasm detection. Section 3 provides the process of the proposed methodology with neat block diagram. Section 4 provides the experimental settings. Section 5 result discusses the performance analysis of the proposed models and existing models. Section 6 provides the concluding remarks.

## 2. Related works

An ensemble classification approach was presented in [11] for sarcasm detection. The model analyzed accuracy, precision, recall but the time complexity was not decreased. Multi-feature fusion approach was developed in [12] to find out the sarcasm data in social media using ML based scheme. However, transfer learning approach was not used to get robust classification results. Multimodal Sentiment and Sarcasm Gradient Cotraining (MSSGC) model was developed in [13] for sarcasm detection using text and data. But, the generalization performance was not attained. A review of Text-based sarcasm detection approaches were introduced in [14] on social networks. An attention model was designed in [15] to achieve emoji mixtured sarcasm detection. A novel detecting scheme for sarcasm in Arabic tweets named as ArSa-Tweet approach was introduced in [16]. It is based on implementing and developing DL models to classify tweets as sarcastic or no sarcastic. A transformer-based approach was introduced in [17] to irony and sarcasm detection. Sarcasm detection using DL was introduced in [18] with contextual features.

New pipeline to augment sarcasm data with Reverse Generative Adversarial Network (RGAN) was introduced in [19]. To detect the sarcasm in Dialogues, explainable artificial intelligence models were designed in [20]. An integrated multilingual BERT (mBERT) embeddings with BiLSTM and Multi-Head Attention (MHA) was designed in [21] to detect the sarcasm. Classical ML techniques, fine-tuning pre-trained language models, and zero-shot inference by multilingual large language models were developed in [22]. Modified Switch Transformer (MST) was introduced in [23] to determine the sarcasm and classify the sentiment and dialect in Arabic text data. A multi-head attention-based bidirectional LSTM was developed in [24] to find the sarcastic data. A hybrid model depended on CNN with multi-head attention and a bi-directional gated recurrent unit (BiGRU) was introduced in [25] to identify the sarcasm.

Image enhancement has also been recognized as an important preprocessing step for improving the performance of deep learning-based visual analysis. Karthik S and Samson Ravindran R [26] proposed a modified residual block-based Cycle Generative Adversarial Network for underwater image enhancement,

demonstrating that enhanced visual representations significantly improve feature extraction and subsequent image classification performance in complex imaging environments.

## 3. Proposed Methodology

Multi-model sarcasm detection is an emerging area of research that aims to improve the understanding and identification of sarcasm in text, and visual formats. Sarcasm, characterized by the use of irony to express a meaning opposite to the truthful interpretation, and includes noteworthy challenges for natural language processing (NLP) due to its reliance on context, tone, and often non-verbal cues. Traditional sarcasm detection methods typically focus on text alone, leveraging linguistic features and sentiment analysis. However, sarcasm often transcends mere words, incorporating facial expressions, and other contextual elements. Multi-model approaches incorporate various modalities such as textual data, and visual content to create a more comprehensive understanding of sarcasm. Accordingly, Pearson Correlated FastText and Bray curtis Deep Recurrent Neural Network (PCFT-BDRNN), Optimized Canonical DenseNet (OP-DenseNet) Model and Aggregated max fusion model are proposed to improve the performance of the sarcasm detection for both using text and images

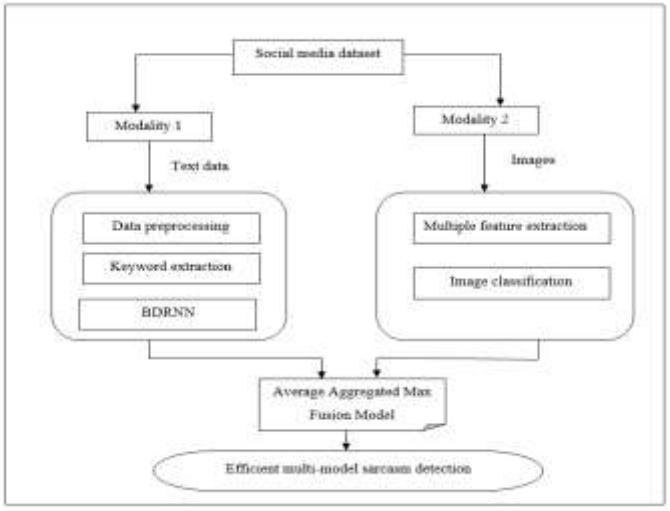


**Figure 2 Block Diagram of Proposed Framework**

Figure 2 demonstrates the block diagram of proposed framework for detecting multi-modal sarcasm. The proposed framework fuses the text and image modality for efficiently detecting the sarcasm with higher accuracy. To detect the sarcasm detection using text data, PCFT-BDRNN method is proposed whereas Optimized Canonical DenseNet Model is employed for detecting sarcasm using images. Later, aggregated max fusion model is applied to fuse both the modalities to achieve robust multi-modal sarcasm detection.

### 3.1 Text based sarcasm detection

Text-based sarcasm detection is a vital spot of research in NLP that addresses the complexities of human language. As digital communication continues to grow, the ability to accurately interpret sarcasm will play an essential role in diverse applications from sentiment analysis to conversational AI. Ongoing research aims to refine detection methods, improve model performance, and ultimately enhance the understanding of nuanced human communication. Many text-based sarcasm unable to accurately understand as the real intention of the sarcastic text might be the opposite of the obvious meaning of the text. Therefore, the original views and thoughts of users are examined by introducing Pearson Correlated FastText and Bray curtis Deep Recurrent Neural Network (PCFT-BDRNN) to detect sarcastic information.

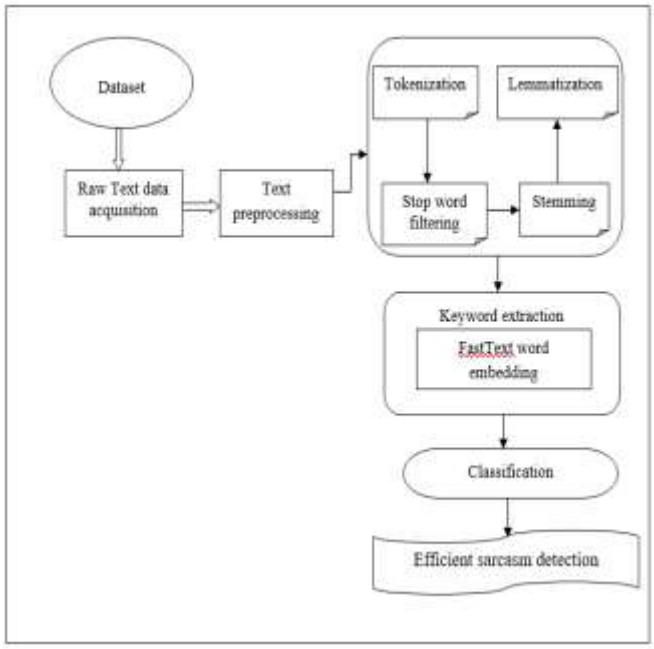


**Figure 3 Block Diagram of Text Data Sarcasm Detection using PCFT-BDRNN**

As shown in above figure, PCFT-BDRNN method is developed with the objective of improving the accuracy of text data sarcasm detection. The proposed network takes Twitter Multi-modal sarcasm detection Dataset as input. With this input, text data pre-processing is performed via tokenization, stop word filtering and word stemming. Then, the word embedding is carried out by using FastText word embedding model where important keywords for sarcasm detection are extracted. Lastly, classification of text data is accomplished using Bray Curtis Deep Recurrent Neural Network for identifying the sarcastic, or non sarcastic text data.

### A. Data collection

Text-based information for sarcasm detection is collected from the Twitter Multi-modal sarcasm detection Dataset. The dataset used in this work is created and shared publicly https://github.com/headacheboy/data-of-multimodal-sarcasm-detection. It includes training set, validation set, and test set. The training set comprises 29,040 tweets, the testing set comprises 2,409 tweets, and the validation set comprises 2,410 tweets. The dataset is split into 80 percent training and 20 percent testing.

### B. Text data preprocessing

The first and foremost process involved in the PCFT-BDRNN method is data preprocessing. During preprocessing, the text data is cleaned and proceed to diminish noise and inappropriate data. It includes three various steps such as tokenization, stemming, and filtering to reduce the time taken in the sarcasm detection. The main aim of preprocessing is to prepare the text data for further analysis with lesser time.

Tokenization is significant step in NLP that involves partitioning of text into minimum units named as tokens. These tokens can be words, phrases, symbols, or other important elements based on the application. Consider the input dataset $D$ and the number of text data is collected from the dataset.

$$d_i = d_1, d_2, d_3, \ldots . d_n \in D \quad (1)$$

Where, '$d_i$' denotes the number of text data '$d_1, d_2, d_3, \ldots . d_n$' gathered from the dataset $D$. Upon collecting the text data, the tokenization process is carried out using Gensim tokenizer. Gensim is an easy and elastic tokenizer where it modifies the raw text into a list of tokens (words or phrases) using punctuation and spaces within the square bracket.

$$(tokenize)d_1 \rightarrow [w_1, w_2, w_3, \ldots . w_x] \quad (2)$$

Where, $d_1$ indicates a text data, $w_1, w_2, w_3, \ldots . w_x$ indicates a tokens i.e., words acquired from the text data.

Stop-word filtering is one of the noteworthy preprocessing techniques to boost the performance of sarcasm detection. The proposed method uses the Pearson Correlation to filter the stop words. Here, the lengths of words are filtered out prior to processing. The most generally used stop words are, “are”, “the”, “a”, “an”, “in, “and”, “our”, “this”, and so on. These words are filtered out from the text data and provide more meaningful information. Pearson correlation is mathematically expressed as follows.

$$\varphi = \frac{x\sum \ w_d * t_f - \sum \ w_d \sum \ w_f}{\sqrt{x\left(\sum \ w_d{}^2 - (\sum \ w_d)^2\right)(x\sum \ w_f{}^2 - (\sum \ w_f)^2)}} \quad (3)$$

Where ‘$w_d$’ denotes the Pearson correlation coefficient, ‘$x$’ indicates the number of words, ‘$w_d$’ refers a words in the text data, and ‘$w_f$’ refers a list of stop words file. The computed Pearson correlation coefficient ‘ $\varphi$’ value is ranges from -1 to +1. The result ‘+1’ refers a higher correlation between the words and word is identified as stop word whereas result ‘0’ indicates no correlation and the word is not a stop word. The identified stop words are filtered from the text data.

Followed by, word stemming is performed. Stemming is the process of taking out the base form (stem) of words by removing affixes from them. In the stemming method, the base word or root form is discovered by cutting off the end or beginning words to find the base word. From that, all the words in the text data are converted into their stem. For example the words ‘playing, information’ are stemmed to ‘play, inform’.

Later, Lemmatization is carried out where it modifies every word into a root word with an appropriate meaning. A lemma is the canonical form of a word it takes the semantic meaning presented in the word. For example: ‘builds, building, built’ is changed into ‘build’, ‘goes, going, went, gone’ are changed into ‘go’.

### C. Word embedding

Word embedding is performed after the text preprocessing to extract more pertinent words from the input text. It characterizes words as uninterrupted vector spaces. Proposed method uses the FastText word embedding to extract the keywords for accurate sarcasm detection. It is an extension of Word2Vec that represents words as n-grams, allowing it to capture subword information and improve performance, especially for morphologically rich languages or rare words.

In FastText, each word is represented as a sum of its n-grams. For example, the word "apple" can be represented by its character n-grams:

For n=3 (trigrams): “<ap”, “app”, “ppl”, “ple” “le>”

The representation is formulated as given below.

$$v_w = \sum_{g \in ngrams} v_g \quad (4)$$

Where ‘$v_w$’ denotes the vector representation of word, and ‘$v_g$’ refers a vector representation of each n-gram. FastText uses a similar objective to Word2Vec, focusing on predicting surrounding context words given a target word using Skip-gram. The main goal of Skip-gram ‘$\delta$’ is to maximize the probability of context words given a target word. It is mathematically given by,

$$\delta = max \sum_{t=1}^{x} \quad \sum_{c \in C_t} \quad Prob(w_c|w_t) \quad (5)$$

Where ‘c’ refers context window size, ‘$C_t$’ refers a context, ‘$x$’ refers a total number of words in the corpus, ‘$w_c$’ refers a context word and ‘$w_t$’ refers a target word. Subsequently, the probability of context word is defined by using softmax function as given below.

$$Prob(w_c|w_t) = \frac{exp\ (v_{w_t}.v_{w_c})}{\sum_{w \in x} \ exp(v_{w_t}.v_w)} \quad (6)$$

Where ‘$x$’ indicates a vocabulary i.e., $w \in \{w_1, w_2, w_3, \ldots . w_x\}$ , ‘$v_{w_t}$’ indicates a vector for the target word and ‘$v_{w_c}$’ indicates a vector for the context word. Based on the probability between context

words and the target keywords, the semantic relationships between words are identified to optimize word representations.

**D. Bray Curtis Deep Recurrent Perceptive Neural Learning**

Followed by the keyword extraction, classification of keywords is performed using Bray Curtis Deep Recurrent Perceptive Neural Learning for sarcasm detection. The designed deep recurrent is an advanced model of conventional RNNs, incorporating multiple layers to detain complex temporal patterns in chronological text data. Deep RNNs are organized into different layers such as such as input layer, multiple hidden recurrent layers and output layer that consist of associated neurons. The outputs from hidden layer are feedback into input of the input layer to get improved results. This feedback facilitates RNNs to store internal state between time steps, enabling them to consider and learn from earlier outputs.

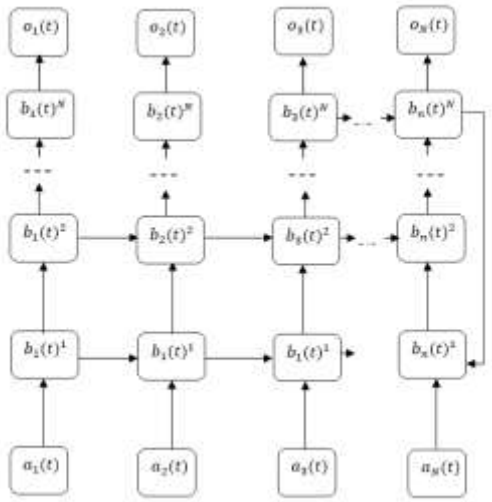

**Figure 4 Structure of Bray Curtis Deep Recurrent Perceptive Neural Learning**

Figure 4 depicts a structural diagram of Bray Curtis Deep Recurrent Perceptive Neural Learning for sarcasm detection. It stacked multiple recurrent layers to deeply analyze the given input and provides accurate classification results. The input layer acquires the input keywords where each input is characterized as a vector. The perceptron obtains extracted keyword vector as input '$k_t$' in the input layer.

$$a(t) = \left[\sum_{k=1}^{K} k_t * \vartheta_i\right] + \beta_i \quad (7)$$

Where, $a(t)$ symbolizes an input function at time 't', $k_t$ symbolizes a keyword vector at time '$t$', $\vartheta_i$ denotes a weight between hidden and input layer, $\beta_i$ symbolizes a bias term in the input layer stored the value is 1. The input is forwarded to the multiple hidden layers to deeply examine the given keyword vector for sarcasm detection.

$$b(t) = \alpha[k_t * \vartheta_i + \vartheta_h * b_{i-1} + \beta_h] \quad (8)$$

Where, '$b(t)$' refers the hidden layer output at time 't', '$\vartheta_h$' refers the weight of hidden layer, $\vartheta_i$ indicates the weight of the input layer, $b_{i-1}$ refers an output of the prior hidden layer, '$\beta_h$' refers a bias term in hidden layer and '$\alpha$' indicates the similarity function to analyze the keyword vectors. Proposed deep learning network uses the Bray Curtis Index for computing the relationship between keyword vectors for classification.

$$\alpha = \frac{\sum |k_t - k_g|}{\sum (k_t + k_g)} \quad (9)$$

Where $\alpha$ denotes the Bray curtis similarity between two keyword vectors, $k_t$ denotes the keyword vector and $k_g$ denotes the testing keyword vector. The results of Bray curtis index provide the output value

from '0' to '1'. The similarity value is given to the next hidden layer where the swish activation function is applied to provide accurate classification results.

$$S_f = k_t * \frac{1}{1+e^{-\theta kg}} \quad (10)$$

Where $S_f$ denotes a swish activation function, and $\theta$ denotes a trainable parameter ( $\theta = 1$). The output of the swish activation function is provided as follows.

$$S_f = \{1\,; \quad sarcastic\ 0\,; \quad non-sarcastic \quad (11)$$

Where $S_f$ returns '1' as output when the similarity output is higher. Otherwise, the activation function returns '0'. Later, Huber Loss is calculated to obtain the final classification results.

$$Hinge_{loss} = \sum \ 1 - T_{S_f}.P_{S_f} \quad (12)$$

Where '$Hinge_{loss}$' denotes a Hinge Loss, '$T_{S_f}$' indicates true results, $P_{S_f}$ denotes predicted classification results. The output of the hidden layer is feedback to the input of the first hidden layer. This process is repeated until the loss gets minimized. If the minimal error is attained, then the results are shown in an output layer as follows,

$$c(t) = (b(t)^{(N)} * \vartheta_o) + \beta_o \quad (13)$$

Where '$c(t)$' refers to a results of output layer at time (t) and it is only based on the hidden state of final $N^{th}$ hidden layer '$b(t)^{(N)}$'. Here, $\vartheta_o$ refers the weight of output layer, '$\beta_o$' refers the bias at output layer. In this way, accurate classification is obtained at the output layer. Based on the classification result, an accurate sarcasm prediction is achieved in PCFT-BDRNN method. The algorithm of Pearson correlated FastText and bray curtis deep recurrent neural network is described as given below.

Algorithm 1 describes the step-by-step process of sarcasm detection using pearson correlated FastText and bray curtis deep recurrent neural network. Initially, text data is collected as input. After collecting text data, data preprocessing process such as Gensim tokenizer, pearson correlation stop-word filtering, stemming and lemmatization are performed to clear format of data. Followed by this, FastText model is employed to identify the keyword vector from the given text data. The extracted keyword vector is given to bray curtis deep recurrent perceptive neural learning. Here, the input layer takes keyword vector as input. Then, the input is forwarded to the multiple hidden layers where the similarity between keyword vector and testing keyword vector are calculated with the help of Bray Curtis Index. The similarity results of bray curtis index is given to the swish activation function. If the activation function provides '1' as output, then the word is classified as sarcasm detection. Otherwise, word is classified as non-sarcasm detection. For each time step, the Hinge loss is computed. If the minimal error is obtained, then the process gets to stop. Otherwise, the output is feedback until the minimum error is found. Finally, the prediction results are acquired at the output layer. This in turn increases the accuracy of sarcasm detection.

**Algorithm 1: Pearson Correlated FastText and Bray curtis Deep Recurrent Neural Network**
**Input**: Datasets '$D$', Number of text data $d_i \in d_1, d_2, d_3 \ldots . d_n$
**Output**: Accurate sarcasm detection
**Begin**
**1.** Gather the input text data $d_i \in d_1, d_2, d_3 \ldots . d_n$
**2.** For each text $d_i$
**3.** Carry out Gensim tokenization to partition the tokens or words using (2)
**4.** For each word '$w_i$'
**5.** Apply Pearson correlation stop-word filtering '$\varphi$' using (3)
**6.** If ($\varphi = 1$) then,
**7.** Word is said to be a stop word
**8.** Filter the stop word
**9.** Else

**10.** Word is not a stop word
**11.** End if
**12.** End for
**13.** Cary out word stemming
**14.** Perform Lemmatization
**15.** Apply Word embedding
**16.** Compute probability of context word '$Prob(w_c|w_t)$' using (6)
**17.** Extract the word embedding vector
**18.** For each embedding vector '$d_i$'
**19.** Formulate the input layer and multiple recurrent hidden layers
**20.** Compute Bray Curtis Index '$\alpha$' using (9)
**21.** Forward the '$\alpha$' score to the next hidden layer
**22.** Apply swish activation function '$S_f$' using (10)
**23.** If ($S_f = 1$) then
**24.** Text data is classified as sarcastic
**25.** Else
**26.** Text data is classified as non sarcastic
**27.** End if
**28.** End for
**29.** For each time step '$t$'
**30.** Measure Hinge loss '$Hinge_{loss}$' using (12)
**31.** If $Hinge_{loss}$ is minimum then
**32.** Obtain the classification results at the output layer using (13)
**33.** Else
**34.** Repeat the step 2
**35.** End if
**36.** End for
**37.** End for
End

### 3.2 Optimized Canonical DenseNet (OC-Densenet) Model

The sarcasm detection is also carried out in proposed method by using images acquired from social media. Detecting sarcasm in images is an intricate task that leverages advances in computer vision and NLP.

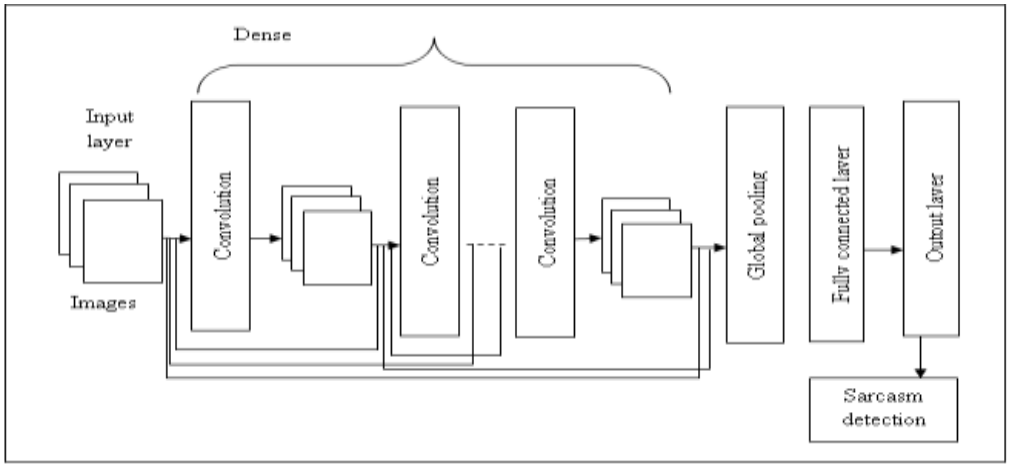


**Figure 5 Structure of OC-DenseNet**

By combining visual features with contextual information, it is possible to create model that can recognize sarcasm. In images, sarcasm might manifest through facial expressions, body language, or visual elements that contrast with the textual content. Images may contain numerous elements that can be interpreted in diverse ways. For instance, a smiling face in a contextually negative situation can suggest sarcasm, but interpreting it accurately requires understanding the underlying message. Understanding sarcasm can enhance user engagement, improve sentiment analysis, and facilitate better communication in automated systems. Therefore, novel model called Optimized Canonical DenseNet (OC-DenseNet) Model is proposed for sarcasm detection. OC-DenseNet is a type of convolutional neural network that uses dense connections between layers through dense blocks. OC-DenseNet architecture offers significant benefits over other CNN including competent feature reuse, and improved performance on difficult tasks. These features make DenseNet a more powerful and flexible choice for a wide range of deep learning applications.As demonstrated in the above figure, OC-DenseNet includes several layers such as input layer, convolutional layers, dense blocks, global average pooling layer, fully connected layer and output layer. Unlike conventional CNN architectures where each layer is associated only to following layers, DenseNet establishes direct connections between all layers within a block. Here, dense block is a key architectural component that train deep networks very effectively. Each layer in a dense block gets input from all earlier layers. This means that the output from every layer is concatenated with the inputs to the next layer, allowing the model to leverage all formerly analyzed features.

The input layer receives images '$M = \{M_1, M_2, \dots, M_n\}$' as input. After the input layer, there is usually a convolutional layer that processes the input image.

$$q_0 = Conv\ (M) \tag{14}$$

Where '$q_0$' denotes an output of convolutional layer. The proposed architecture includes several dense blocks, each containing multiple convolutional layers. Each layer within a dense block receives input from all preceding layers. The convolutional operation is carried out to extract the more pertinent features such as texture, shape, color and boundary from the input images. These extracted features are used for sarcasm detection.

First, proposed model extracts the texture features from the images using Concordance index. It involves dependent variables (i.e., image pixels) and predictor variables (i.e., the mean of the pixel intensity). The texture feature gives information about the spatial representation.

$$Texture_{\ F} = 2 * \frac{\left[\frac{1}{n}(\rho_i - \rho_j)(\rho_i - \rho_j)\right]}{\sigma_i \sigma_j} \tag{15}$$

Where $Texture_{\ F}$ denotes texture, the pixel $\rho_i$ and its neighboring pixels ' $\rho_j$' based on the mean ( $\mu_i$) of the pixel, and the deviation of the pixels '$\sigma_i$', and deviation of the pixels ' $\sigma_j$'.

The shape features of the object are extracted by a contour in which the center of the image is denoted by$(0, 0)$. The gray level intensity contrast of the image is measured as the difference between the pixel $(\rho_i)$ and their neighboring pixels $(\rho_j)$ in the set of the pixel.

$$G_{Inten} = \sum_i \quad \sum_j \quad \left|\rho_i - \rho_j\right|^2 \tag{16}$$

Where $G_{Inten}$ denotes a gray level intensity contrast. The color features are extracted by transferring the RGB image into HSV (hue, saturation, value) color spaces. In addition, deviation between the pixels obtainable over an input image is denoted through the standard deviation feature and it is expressed as given below.

$$\sigma = \sqrt{\frac{1}{z}\sum_{r=1}^{m} \quad \sum_{c=1}^{n} \quad (k[r, c] - \mu)^2} \tag{17}$$

Where $z$ is a number of pixels of an image and $k(r, c)$ is a value of the equivalent pixel at row $r$ and column $c$.

Later, graph based edge detection is performed to detect the edges of an image object using Graph-based edge detection. Graph-based methods treat the image as a graph, allowing for adaptive edge detection. Each pixel is a node, with edges defined by pixel similarity as follows.

$$\rho_{sim} = exp\left(-\frac{\|\rho_i - \rho_j\|^2}{s_{factor}}\right) \quad (18)$$

Where '$\rho_{sim}$' refers a similarity between pixels $\rho_i$ and $\rho_j$ and '$s_{factor}$' is a scaling factor. The minimal similarity between the pixels is identified as edges of the object.

The results of extracted features are given to the global average pooling layer where the spatial dimensions of the input feature maps are optimized. Before the output layer, DenseNet often includes a global average pooling layer that reduces the spatial dimensions of the feature maps to a single vector, effectively summarizing the learned features before feeding them into the output layer.

Consider 'F' input feature map of size $h \times \omega$ and it is mathematically given by

$$F = \begin{bmatrix} f_{1,1} & \cdots & f_{1,\omega} \\ \vdots & \ddots & \vdots \\ f_{h,1} & \cdots & f_{h,\omega} \end{bmatrix} \quad (19)$$

The size of pooling is $2 * 2$ and Stride is considered as '2'. Then, the output size after pooling can be calculated as:

$$y_h = \left(\frac{h-l}{s} + 1\right) \quad (20)$$

$$y_\omega = \left(\frac{h-l}{s} + 1\right) \quad (21)$$

Where '$l$' refers a pool size and the pooled output $L$ is computed as:

$$L_{i,j} = max\{FM_{m,n} | m \in (i.s, i.s + l), n \in (j.s, j.s + l)\} \quad (22)$$

Where, $L_{i,j}$ represents the output of the global average pooling operation at position $(i, j)$ in the output feature map, $FM_{m,n}$ indicates the extracted feature values in the input feature map, and $max$ indicates the process of higher value specified set of values of input feature map.

Lastly, fully connected layers are employed after the global average pooling to classify the extracted features for sarcasm detection. This layer maps the pooled feature vector to the output classes. Fully connected layers use the Canonical variate analysis to classify the input extracted feature vector. It computes the relationship between training features (i.e. extracted features) and testing features and it is mathematically expressed as follows.

$$C_{VA} = \frac{\sum (\varepsilon f_i - \varepsilon f_i')(\gamma f_i - \gamma f_i')}{n} \quad (23)$$

Where '$C_{VA}$' point outs a Canonical variate analysis, '$\varepsilon f_i$' denotes an extracted feature map, '$\varepsilon f_i'$' refers the mean value of the extracted feature, '$\gamma f_i$' refers to a testing feature value, '$\gamma f_i'$' refers to a testing feature mean value and '$n$' point outs a total number of the features extracted. Subsequently, Rectified Linear Unit (ReLU) activation function is used in the network to examine the results obtained from Canonical variates. Thus, output of ReLU activation function is given below.

$$ReLU = \{1 \quad if \ C_{VA} > 0 \quad 0 \quad if \ C_{VA} \leq 0 \quad (24)$$

Where '$ReLU\ (C_{VA})$' indicates the ReLU activation function. If the correlation between features are higher (i.e., $C_{VA} > 0$ ), the ReLU returns '1' as output and the image is classified as sarcastic. Otherwise, the image is classified as non-sarcastic. With this, the performance of sarcasm detection is improved with better accuracy. Algorithm of OC-DenseNet is described as follows.

**Algorithm 2: Optimized Canonical DenseNet Model**

**Input:** Dataset, images $M = \{M_1, M_2, \ldots, M_n\}$

**Output:** Efficient sarcasm detection

**Begin**

**1.** Initialize number of images
**2.** For each image '$M_i$'
**3.** // Convolutional layers
**4.** Perform Concordance texture feature extraction using (15)
**5.** Extract intensity, standard deviation features using (16) and (17)

**6.** Carry out Graph-based edge detection using (18)
**7.** End for
**8.** For each extracted feature
**9.** //Global average pooling layer
**10.** Derive the input feature map
**11.** Get pooled output
**12.** Optimize the spatial dimensions of the input image
**13.** End for
**14.** //Fully connected layer
**15.** For each extracted feature vector
**16.** For each testing feature vector
**17.** Perform Canonical variate analysis using (23)
**18.** Apply Rectified Linear Unit (ReLU) activation using (24)
**19.** If ($ReLU = 1$) then,
**20.** Image is labeled as sarcastic
**21.** Else
**22.** Image is labeled as non sarcastic
**23.** End if
**24.** End for
**25.** Get final classification results (output layer)
**26.** End for
End

Algorithm 2 illustrates the process of sarcasm detection using Optimized Canonical DenseNet Model. With the objective of achieving higher accurate sarcasm detection, DenseNet model comprises multiple layers. The set of images are fed to the input layer where weights and biases are assigned randomly to the given input. After that, the initial convolutional layer is employed to extract the basic features such as texture, color, intensity, and standard deviation features in the input images. The extracted primary features from the images are forwarded to the dense block. Dense block in the network includes multiple convolutional layers to extract edge feature using Graph-based edge detection. Within a dense block, each layer gets input from all preceding layers, allowing for feature reuse. Exacted features are send to the global average pooling layer where the spatial dimensions of the images is reduced. The results are passed to the fully connected layer for classifying the features with testing features using ReLU activation function. Based on the activation function, classification results of images are acquired at the output layer. From this, the efficient sarcasm detection is achieved with better accuracy.

### 3.3 Aggregated Max Fusion Model

Proposed method performs multi-modal fusion process to enhance the overall performance. Multimodal sarcasm detection represents to the procedure of detecting sarcastic expressions through examining the multiple types of text and images simultaneously. Proposed method uses the Aggregated Max Fusion Model to fuse the score values obtained from multimodal sarcasm detection (including text data and image).

Let assume score values obtained from text data classification output is '$Q_{td}$' and score values obtained from image classification output is '$Q_{im}$'. Then, assign weights to each modality depended on their performance, allowing better-performing modalities to have more influence. Thus, the weighted aggregation is performed as follows.

$$Q = w_{td}.Q_{td} + w_{im}.Q_{im} \quad (25)$$

Where '$Q$' is fusion output, '$w_{td}$' refers a weight of text data, and '$w_{im}$' refers a weight of image. From that, the maximum score among the modalities for each class, emphasizing the strongest prediction. If the weight of '$w_{td}$' and '$w_{im}$' are maximum, then the sarcasm is detected from the multi-modal input.

$$Q_{final} = max(Q_{td}, Q_{im}) \quad (26)$$

From the above equation, the highest score value of multi-modal inputs indicates the sarcastic whereas lowest value indicates a non sarcastic. From this, the accurate sarcasm detection is achieved in proposed method. Algorithm of aggregated max fusion model for sarcasm detection is described below.

**Algorithm 3: Aggregated max fusion model**

**Input:** Text data classification output '$Q_{td}$', Image classification output '$Q_{im}$'
**Output:** Multimodal Sarcasm Detection
**Begin**
1. **For** each Text data classification output '$Q_{td}$'
2. **For** each Image classification output '$Q_{im}$'
3. Compute weighted aggregation '$Q$' using (25)
4. Identify the maximum score values from $Q_{td}$ and $Q_{im}$ using (26)
5. Predict the sarcasm

**End**

As shown in the above algorithm, aggregated max fusion model is carried out with the objective of improving the sarcasm detection through fusing the output from multi-modals. To do this, initially weighted aggregation is performed to aggregate the score values from text classifier and image classifier. The aggregated max fusion model is used to fuse the maximum score values obtained from classifiers for sarcasm detection.

### 3.4 Governance Implications

The increasing use of digital platforms by Local Self-Government (LSG) institutions has transformed citizen engagement and public service management. Municipal corporations, panchayats, and urban local bodies actively utilize social media, mobile applications, and online grievance portals to communicate with citizens and gather feedback on civic services. These platforms generate large volumes of multimodal data containing textual comments and images that reflect public perceptions of governance. However, citizens often express dissatisfaction through sarcasm or irony, making it difficult for conventional sentiment analysis techniques to accurately interpret their true intentions. Consequently, important public concerns may remain unidentified, affecting timely administrative action and service improvement.

The proposed multimodal sarcasm detection framework provides an intelligent decision-support tool for Local Self-Government authorities by accurately identifying sarcastic expressions from textual and visual content. The framework assists administrators in monitoring public opinion, prioritizing citizen grievances, evaluating municipal services such as waste management, water supply, road maintenance, public transportation, and taxation, and improving communication with residents. Furthermore, it supports evidence-based policymaking by providing reliable insights into citizen satisfaction and service quality. During emergency situations, the framework can also facilitate effective disaster communication by distinguishing genuine concerns from sarcastic or misleading responses. Therefore, integrating the proposed framework into digital governance platforms enhances transparency, accountability, citizen participation, and responsive local administration while supporting sustainable and data-driven Local Self-Government.

## 4. Experimental Settings

An experimental evaluation of the proposed methods such as PCFT-BDRNN, OP-DenseNet and Aggregated max fusion model are implemented in python. The performance of sarcasm detection using text and image data are performed using Twitter Multi-modal sarcasm detection Dataset https://github.com/headacheboy/data-of-multimodal-sarcasm-detection.The dataset comprises a collection of English tweets includes a picture and hashtags to find the sarcastic or non sarcastic data. To carry outs a experimentation, 1000 to 10000 text and images are considered.

## 5. Performance Analysis

In this section, the performance analysis of the PCFT-BDRNN, OP-DenseNet and Aggregated max fusion model are analyzed and compared with existing Cue learning [1], KnowleNet [2] are compared with metrics such as accuracy, precision, recall, and sarcasm detection time.

- **Accuracy:** It is defined as the ratio of correctly classified texts or images as sarcastic or non-sarcastic to the overall of inputs involved in the experiments. It is typically evaluated as follows,

$$Accuracy = \left(\frac{TP+TN}{TP+TN+FP+FN}\right) * 100 \quad (27)$$

Where '$TP$' is a true positive rate (correct detection of sarcasm or non sarcasm), '$TN$' is a true negative rate (non-sarcastic instances correctly identified as non-sarcastic), '$FP$' is a false positive rate (incorrect detection of non sarcasm as sarcasm) and '$FN$' false negative rate (incorrect detection of sarcasm as non sarcasm).

- **Precision: It computes the** ratio of true positive predictions among all the positive predictions. Precision is calculated as follows**,**

$$Precision = \left(\frac{TP}{TP+FP}\right) * 100 \quad (28)$$

Precision is calculated in percentage (%).

- **Recall**: It measures an ability to sense all relevant positive samples in a dataset.

$$Recall = \left(\frac{TP}{TP+FN}\right) * 100 \quad (29)$$

The results of recall also computed in percentage (%).

- **Sarcasm detection time:** It is estimated as the amount of time used through the model to perform sarcasm detection using text or image.

$$SDT = \sum_{i=1}^{n} S_i * Time\ [(SD)] \quad (30)$$

Where '$SDT$' refers a sarcasm detection time, '$n$' refers the number of samples '$S$', $Time\ (SD)$ refers a sarcasm detection time. It is determined in milliseconds (ms).

### 5.1 Results sarcasm detection using text data

The result of sarcasm detection is analyzed using proposed PCFT-BDRNN with existing Cue learning [1], KnowleNet [2] via above mentioned metrics.

**Table 1 Comparison of Accuracy**

| Number of texts | Accuracy (%) | | |
|---|---|---|---|
| | PCFT-BDRNN | Cue learning [1] | KnowleNet [2] |
| **1000** | 94.50 | 90.50 | 88.00 |
| **2000** | 93.45 | 89.56 | 86.45 |
| **3000** | 94.05 | 88.45 | 85.74 |
| **4000** | 93.46 | 90.45 | 88.04 |
| **5000** | 94.00 | 89.74 | 87.04 |
| **6000** | 94.74 | 88.05 | 86.66 |
| **7000** | 93.78 | 89.74 | 86.45 |
| **8000** | 93.74 | 87.05 | 85.05 |
| **9000** | 94.78 | 88.74 | 86.41 |
| **10000** | 94.02 | 88.36 | 86.96 |

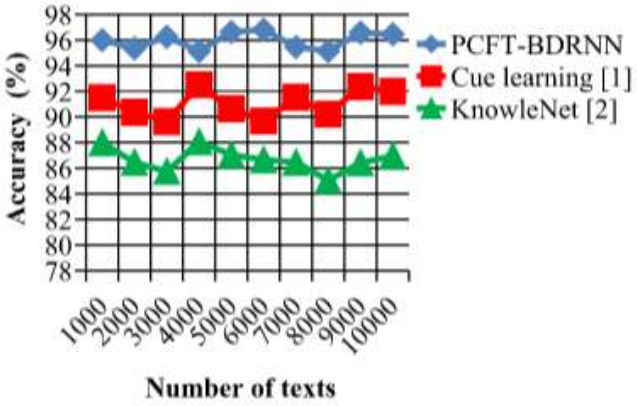


**Figure 6 Graphical representation of Accuracy using texts**

Figure 6 describes the performance results of accuracy for sarcasm detection using text data considered in the ranges from 1000 to 10000. The performance of sarcasm detection accuracy is computed for three methods such as PCFT-BDRNN with existing Cue learning [1], and KnowleNet [2]. From the graphical representation, the accuracy is improved for proposed PCFT-BDRNN model during the sarcasm detection using text data than the existing Cue learning [1], and KnowleNet [2]. This can be proved using statistical calculation. With the input of 1000 text data, the accuracy is obtained as 96% for proposed PCFT-BDRNN model whereas 91.5% and 88% of accuracy is measured for conventional [1], and [2] respectively. Likewise, the accuracy is computed up to 10000 text data.

**Table 2 Comparison of Precision**

| **Number of texts** | **Precision (%)** | | |
|---|---|---|---|
| | **PCFT-BDRNN** | **Cue learning [1]** | **KnowleNet [2]** |
| **1000** | 97.07 | 94.11 | 91.56 |
| **2000** | 96.25 | 93.63 | 89.56 |
| **3000** | 96.45 | 92.63 | 88.45 |
| **4000** | 97.01 | 92.25 | 87.05 |
| **5000** | 96.11 | 91.32 | 87.45 |
| **6000** | 96.25 | 91.25 | 88.65 |
| **7000** | 95.63 | 91.63 | 87.48 |
| **8000** | 95.11 | 92.36 | 88.32 |
| **9000** | 96 | 92.82 | 89.05 |
| **10000** | 95.63 | 91.36 | 87.45 |

The output results are compared with existing methods. The main reason behind the accuracy improvement is achieved using by performing FastText word embedding where the pertinent keywords are extracted and it is given to the deep RNN for classification. In the classification process, the Bray Curtis index is computes the similarity between training and testing keyword vectors. According to the similarity measure, the sarcastic and non sarcastic text data is detected with better accuracy. Thus, the results of accuracy of sarcasm detection using proposed PCFT-BDRNN model is improved by 5% and 11% as compared to existing Cue learning [1], and KnowleNet [2] respectively.

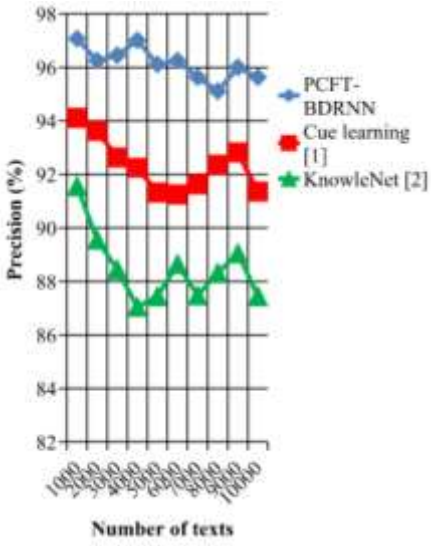


**Figure 7 comparison of precision for text data using proposed and existing methods**

Figure 7 demonstrates the comparison of precision for text data using proposed and existing methods. As shown in above figure, X-axis and Y-axis show the number of text data and precision results for three methods. The blue, red, and green color lines represents the performance of precision using proposed PCFT-BDRNN with existing Cue learning [1], and KnowleNet [2] respectively. The graphical results indicate that the performance of precision using PCFT-BDRNN is found to be increased than the other methods. In the first run, the precision of PCFT-BDRNN is measured as 97.07% and Cue learning [1], and KnowleNet [2] are measured as 94.11%, and 91.56% respectively. In all the runs, the performance of precision is better in PCFT-BDRNN. This significant improvement is attained using bray curtis deep recurrent perceptive neural learning. The designed network deeply analyzes the given input of pertinent words and provides the classification output using swish activation function. This leads to enhance the precision in PCFT-BDRNN by 4% and 9% compared to existing Cue learning [1], and KnowleNet [2] respectively.

**Table 3 Comparison of Recall**

| **Number of texts** | **Recall (%)** | | |
|---|---|---|---|
| | **PCFT-BDRNN** | **Cue learning [1]** | **KnowleNet [2]** |
| **1000** | 98.22 | 95.59 | 93.82 |
| **2000** | 97.52 | 95.11 | 92.05 |
| **3000** | 96.64 | 93.63 | 90.45 |
| **4000** | 98.25 | 92.54 | 89.64 |
| **5000** | 97.25 | 91.52 | 88.45 |
| **6000** | 96.41 | 94.63 | 87.96 |
| **7000** | 97.52 | 93.41 | 90.45 |
| **8000** | 97.11 | 92.47 | 89.41 |
| **9000** | 96.52 | 91.58 | 87.65 |
| **10000** | 97.25 | 92.46 | 86.05 |

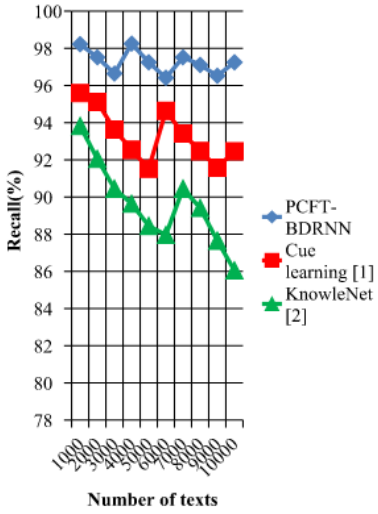


**Figure 8 Graphical representation of Recall using texts**

Figure 8 illustrates the performance effects of recall for text data using proposed PCFT-BDRNN with existing Cue learning [1], and KnowleNet [2]. Horizontal axis indicates number of texts collected from the dataset, ranging from 1000 to 10000 whereas vertical axis denotes performance of recall. As depicted in Figure 7, the results indicate that the recall rate achieved with the PCFT-BDRNN is relatively higher compared to existing methods [1] and [2]. This conclusion is supported by statistical evaluation. The input of 1000 texts of proposed PCFT-BDRNN is found to be 98.22%, while conventional methods [1] and [2] achieved recall 95.59%, and 93.82%. Thus, the average comparison of recall using PCFT-BDRNN is enhanced by 4% and 9% than the [1] and [2] respectively. This improvement is achieved by applying bray curtis deep recurrent perceptive neural learning to detect the sarcasm, and on sarcasm classes correctly.

**Table 4 Comparison of Sarcasm detection time**

| Number of texts | Sarcasm detection time (ms) | | |
|---|---|---|---|
| | **PCFT-BDRNN** | **Cue learning [1]** | **KnowleNet [2]** |
| **1000** | 52 | 60 | 72 |
| **2000** | 58.2 | 68.4 | 80.6 |
| **3000** | 62 | 75.2 | 88.5 |
| **4000** | 70.1 | 80.2 | 95.4 |
| **5000** | 75 | 87.6 | 105.8 |
| **6000** | 86.6 | 104.6 | 121.5 |
| **7000** | 92 | 110.5 | 129.6 |
| **8000** | 102.5 | 120.8 | 138.5 |
| **9000** | 118.4 | 127.4 | 148.5 |
| **10000** | 128.1 | 140.6 | 162.4 |

Figure 9 presents a graphical representation of the sarcasm detection time using three methods namely PCFT-BDRNN with existing methods. The graph demonstrates that sarcasm detection time for every three methods increases as number of texts used in the experiments increases. This is because a larger number of texts take more time to analyze for sarcasm classification. In experiments conducted with 1000 texts, the time consumed for sarcasm detection 52ms, 60ms and 72ms using PCFT-BDRNN, Cue learning [1], and KnowleNet [2].

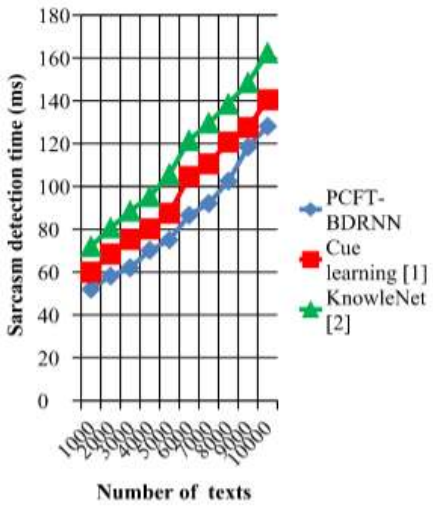


**Figure 9 Graphical representation of Sarcasm detection time using texts**

Overall, outcomes denote the PCFT-BDRN significantly reduces time consumption of sarcasm detection by 14% and 27% compared to methods [1] and [2] respectively. The lesser time consumption is attained by using text preprocessing in PCFT-BDRN. During this process, the Gensim tokenizer, stop-word filtering, stemming and lemmatization are carried out. Also, the word embedding is performed to minimize the dimensionality of dataset. This in turns, the sarcasm detection time is reduced in PCFT-BDRN than the other methods.

### 5.2 Results sarcasm detection using images

The result of sarcasm detection is analyzed using images for proposed OP-DenseNet with existing Cue learning [1], and KnowleNet [2].

**Table 5 Comparison of Accuracy**

| Number of Images | Accuracy (%) | | |
|---|---|---|---|
| | OP-DenseNet | Cue learning [1] | KnowleNet [2] |
| **1000** | 94.8 | 90 | 87 |
| **2000** | 94.83 | 89.68 | 86.65 |
| **3000** | 95.23 | 89.14 | 85.05 |
| **4000** | 94.82 | 90.63 | 87.45 |
| **5000** | 93.58 | 89.36 | 86.52 |
| **6000** | 95.45 | 89 | 86.41 |
| **7000** | 95.21 | 91.25 | 87.63 |
| **8000** | 96.24 | 92.36 | 88.02 |
| **9000** | 95.42 | 91.36 | 86.41 |
| **10000** | 95.11 | 90.34 | 85.04 |

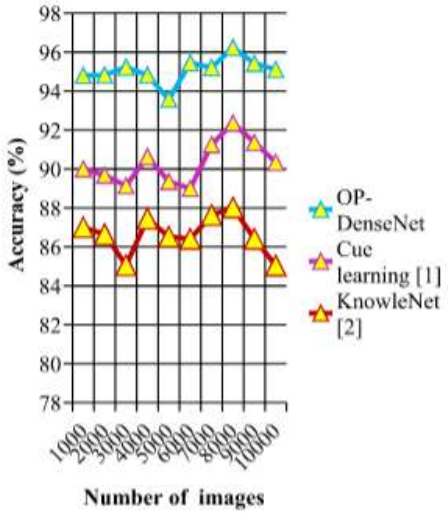


**Figure 10 Graphical representation of accuracy using images**

Figure 10 depicts the performance analysis of accuracy using images collected from the given dataset. The results of accuracy using proposed OP-DenseNet is computed and compared with existing Cue learning [1], and KnowleNet [2] for validation. In the above figure, '$x$' axis indicates number of images and '$y$' axis denotes sarcasm detection accuracy for three methods. Overall performance outcome indicates that the OP-DenseNet enhances accuracy compared to other techniques.

**Table 6 Comparison of Precision**

| Number of Images | Precision (%) | | |
|---|---|---|---|
| | **OP-DenseNet** | **Cue learning [1]** | **KnowleNet [2]** |
| **1000** | 97.21 | 94.26 | 91.66 |
| **2000** | 96.25 | 93.52 | 90.56 |
| **3000** | 95.63 | 92.14 | 89.05 |
| **4000** | 96.69 | 93.25 | 88.04 |
| **5000** | 97.85 | 92.41 | 87.56 |
| **6000** | 96.26 | 93.62 | 88.02 |
| **7000** | 98 | 93.52 | 89.45 |
| **8000** | 97.52 | 92.63 | 87.45 |
| **9000** | 97.82 | 93.41 | 88.44 |
| **10000** | 97.25 | 93.24 | 87.02 |

This improvement is due to the application of the Optimized Canonical DenseNet Model. The designed model comprises numerous layers where dense block is used to connect the each layer input with preceding layer output. In the DenseNet Model, texture, intensity, edge features are extracted from the input image. These extracted features are classified using ReLU activation function to identify the sarcastic images. With this, the accuracy of sarcasm detection is improved by 5% and 10% as compared to existing Cue learning [1], and KnowleNet [2] respectively.

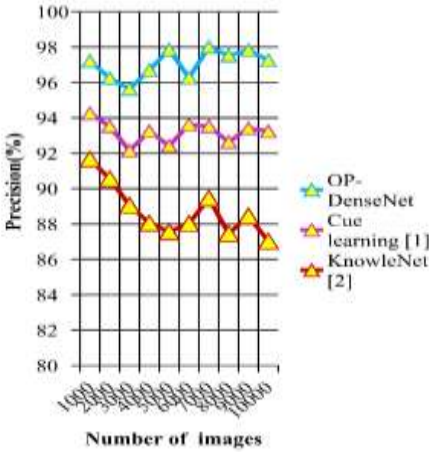


**Figure 11 Graphical representations of precision using images**

Figure 11 depict the experimental results of precision using images for OP-DenseNet, existing Cue learning [1], and KnowleNet [2]. The precision is measured with respect to a number of images in the ranges from 1000 to 10000. The obtained results reveal that the precision is considerably increased by using the OP-DenseNet than the other two deep learning methods.

**Table 7 Comparison of Recall**

| Number of Images | Recall (%) | | |
|---|---|---|---|
| | OP-DenseNet | Cue learning [1] | KnowleNet [2] |
| **1000** | 96.24 | 93.08 | 91.61 |
| **2000** | 95.17 | 92.82 | 88.89 |
| **3000** | 94.63 | 91.63 | 87.45 |
| **4000** | 96.05 | 91.92 | 88.05 |
| **5000** | 95.68 | 92.63 | 87.41 |
| **6000** | 95.14 | 91.82 | 87.33 |
| **7000** | 96.41 | 92.67 | 88.44 |
| **8000** | 96.82 | 91.47 | 86.05 |
| **9000** | 95.36 | 92.63 | 87.44 |
| **10000** | 95.41 | 93.41 | 89.46 |

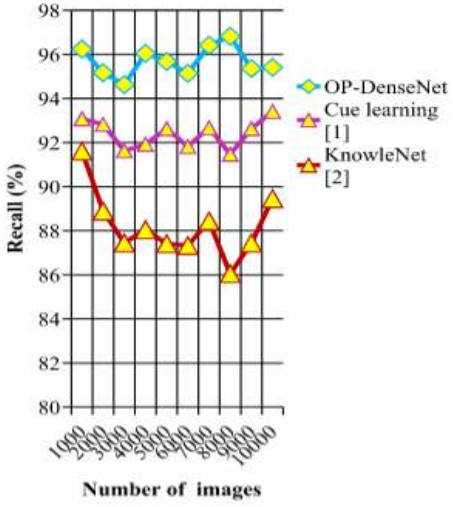


**Figure 12 Graphical representation of recall using images**

Let us consider the 1000 images to perform experiments. The precision of proposed OP-DenseNet is achieved as 97.21%, Cue learning [1] is achieved as 94.26% and KnowleNet [2] is achieved as 91.66% respectively. In the same way, the remaining results are observed for each method with different counts of input images. The average of ten results indicates that the OP-DenseNet increases the performance of the precision by 4% and 9% when compared to existing [1] [2] respectively. This is because of using Optimized Canonical DenseNet Model for computing the correlation between extracted feature map to the testing feature map. According to the correlation analysis, the accurate sarcasm detection is achieved in OP-DenseNet. As a result, the precision is improved in the OP-DenseNet than the existing methods.

Figure 12 shows the evaluation results of recall using images for three various methods such as for OP-DenseNet, existing Cue learning [1], and KnowleNet [2]. The number of image is taken in the ranges from 1000 to 10000 for conducting the experimental evaluation. The observed results indicate that the recall using proposed OP-DenseNet is fund to be higher when compared to other methods. The results of recall using OP-DenseNet are obtained as 96.24% whereas the conventional [1] and [2] obtains 93.08% and 91.61% of recall while processing 1000 images for sarcasm detection. Similarly, recall value is analyzed up to 10000 images. The obtained results indicates that the recall of proposed OP-DenseNet is enhanced by 4% and 9% than the Cue learning [1], and KnowleNet [2] respectively. The enhancement in the recall is attained by extracting the significant features from the images using convolutional layers. Also, the spatial dimensionality of the images is reduced through the pooling layers. Later, the fully connected layer computes association between feature map for sarcasm image classification. This in turns, the recall value is improved in OP-DenseNet than the other methods.

**Table 8 Comparison of sarcasm detection time**

| Number of Images | Sarcasm detection time (ms) | | |
|---|---|---|---|
| | **OP-DenseNet** | **Cue learning [1]** | **KnowleNet [2]** |
| **1000** | 58 | 69 | 78 |
| **2000** | 63.6 | 73.5 | 82.6 |
| **3000** | 70.2 | 80.4 | 90.5 |
| **4000** | 77.5 | 88.5 | 98.6 |
| **5000** | 85.4 | 103.6 | 113.5 |
| **6000** | 100.3 | 113.2 | 122.7 |
| **7000** | 108.2 | 122.3 | 132 |
| **8000** | 120.6 | 130.5 | 141.5 |
| **9000** | 128.5 | 141.6 | 148.6 |
| **10000** | 137.5 | 155 | 162 |

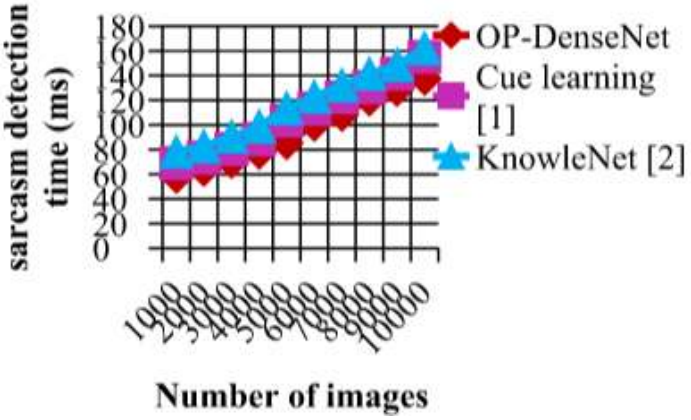


**Figure 13 Graphical representation of sarcasm detection time using images**

Figure 13 depicts the performance results of sarcasm detection time using three different methods. As shown in the above figure, the number of images is taken as input in the x-axis and the results of sarcasm detection time are acquired at the 'y' axis. The above graphical result shows that the sarcasm detection time is said to be improved linearly for all the methods. But relatively, sarcasm detection time is found to be lower in proposed OP-DenseNet than the existing Cue learning [1] and KnowleNet [2] with the application of feature extraction process. OP-DenseNet includes multiple convolutional layers where the features such as texture, color, intensity, standard deviation features and edge features in the input images from the images extracted. Then, the dimensionality of images is decreased in the global average pooling layer. On the contrary processing entire images, processing of spatial dimension reduced images for classification reduces the time needed for sarcasm detection. Therefore, the sarcasm detection time of OP-DenseNet model is reduced by 12% and 20% as compared to existing Cue learning [1],and KnowleNet [2] respectively.

### 5.3 Results sarcasm detection using fusion of text and images

**Table 9 Overall performance results of fusion using text and images**

| Methods | Accuracy (%) | Precision (%) | Recall (%) | Sarcasm detection time (ms) |
|---|---|---|---|---|
| **Aggregated max fusion model** | 97.25 | 95.82 | 97.52 | 145 |
| **Cue learning [1]** | 91.63 | 88.52 | 90.63 | 162 |
| **KnowleNet [2]** | 88.74 | 84.75 | 85.75 | 175 |

The result of sarcasm detection is analyzed using fusion of text and images are analyzed in this section using fusion models for sarcasm detection.

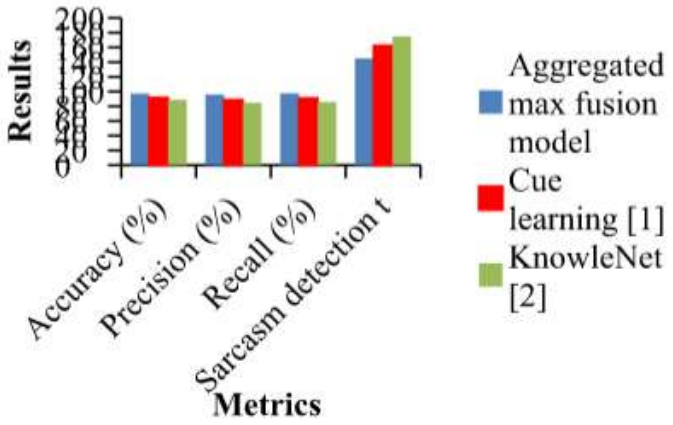


**Figure 14 Overall performance graphical results using text and images**

Figure 14 illustrates the overall performance outcomes of accuracy, precision, recall and sarcasm detection time for three methods such as aggregated max fusion model, Cue learning [1] and KnowleNet [2]. The above said metrics result are enhanced by using aggregated max fusion model than the other

methods. The major reason behind the enhancement is achieved by using fusion model where it fuses the results acquired from both text and image. If the results fusion is maximum, then the max fusion identifies the sarcastic. Otherwise, non sarcastic results are acquired. As a result, overall performance of sarcastic detection is improved using aggregated max fusion model than the other methods. The results of sarcasm detection accuracy using proposed Average aggregated max fusion model is improved by 6% and 10% compared to the existing methods [1] and [2]. The results of precision using Average aggregated max fusion model increased by 8% and 13% than the existing methods [1] and [2]. Also, Average aggregated max fusion model improved the recall by 8% and 14% compared to [1] and [2]. Additionally, sarcasm detection time of Average aggregated max fusion model is decreased by 10% and 17% compared to the existing methods.

## 5. Conclusion

This paper introduces a novel deep learning model for multi-modal sarcasm detection via analyzing the overview of thoughts of users in social media. Pearson Correlated FastText and Bray curtis Deep Recurrent Neural Network (PCFT-BDRNN) model is initially proposed for sarcasm detection using text data. Subsequently, Optimized Canonical DenseNet (OP-DenseNet) Model is developed specifically for images to detect sarcastic images. Later, aggregated max fusion model is implemented to improve the overall performance of the sarcasm detection for in terms of text and images. Experiments are conducted for analyzing the performance of the proposed fusion model with conventional Cue learning and KnowleNet methods. Proposed aggregated max fusion model achieves better results on publicly available benchmark dataset demonstrating its superior performance in multimodal sarcasm detection in terms of accuracy, precision, recall, F1-score, and sarcasm detection time for multiple text and images. The obtained performance of proposed fusion model enhances the accuracy, precision and recall for sarcasm detection with less time than the state-of-the-art deep learning models. Beyond improving sarcasm detection accuracy, the proposed framework has significant implications for digital governance. Local Self-Government institutions can employ the developed model to monitor citizen engagement, analyse public perception regarding municipal services, detect emerging issues from social media, and support evidence-based policy formulation. Therefore, the proposed framework represents an intelligent decision-support tool for modern Local Self-Government.

**Data availability** All data generated or analyzed during this study are included in this Published article.

**Conflict of Interest** The authors declare that they have no conflicts of interest relevant to the content of this manuscript.

**Funding** No Funding was received for conducting this study.

## References

[1] Ming Lu, Zhiqiang Dong, Litian Zhang, Ziming Guo, and Xiaoming Zhang, “A Multi-Modal Sarcasm Detection Model Based on Cue Learning”, Research Square, Springer, 2024, Pages 1-16

[2] Tan Yue, Rui Mao, Heng Wang, Zonghai Hu, Erik Cambria, “KnowleNet: Knowledge fusion network for multimodal sarcasm detection”, Information Fusion, Elsevier, Volume 100, 2023, Pages 1-10. https://doi.org/10.1016/j.inffus.2023.101921

[3] Mohamed A, Galal, Ahmed Hassan Yousef , Hala H. Zayed, Walaa Medhat, “Arabic sarcasm detection: An enhanced fine-tuned language model approach”, Ain Shams Engineering Journal, Elsevier, 2024, Pages 1-14

[4] Rishabh Misra, Prahal Arora, “Sarcasm detection using news headlines dataset”, AI Open, Elsevier, 2023, Pages 13-18

[5] Daniel Šandor and Marina Bagi Babac, “Sarcasm detection in online comments using machine learning”, Information Discovery and Delivery, Volume 52 Issue 2, 2024, Pages 213-226

[6] Alaa M. A. Barhoom1, Bassem S. Abunasser, Samy S. Abu-Naser, “Sarcasm Detection in Headline News using Machine and Deep Learning Algorithms”, International Journal of Engineering and Information Systems (IJEAIS), Volume. 6 Issue 4, 2022, Pages 66-73

[7] Dnyaneshwar Madhukar Bavkar, Ramgopal Kashyap, and Vaishali Khairnar, "Multimodal Sarcasm Detection via Hybrid Classifier with Optimistic Logic", Journal Of Telecommunications And Information Technology (JTIT), 2022, Pages 97-114
[8] Oxana Vitman, Yevhen Kostiuk, Grigori Sidorov, Alexander Gelbukh, "Sarcasm Detection Framework Using Context, Emotion and Sentiment Features", Expert Systems with Applications, Elsevier, Volume 234, 2023, Pages 1-20
[9] Ananya Pandey, Dinesh Kumar Vishwakarma, "VyAnG-Net: A Novel Multi-Modal Sarcasm Recognition Model by Uncovering Visual, Acoustic and Glossary Features", Computer Vision and Pattern Recognition, 2024, Pages 1-24
[10] Ashraf Kamal and Muhammad Abulaish, "CAT-BiGRU: Convolution and Attention with Bi-Directional Gated Recurrent Unit for Self-Deprecating Sarcasm Detection", Cognitive Computation, Springer, 2022, Pages 91-109
[11] Jyoti Godara, Isha Batra, Rajni Aron, and Mohammad Shabaz, "Ensemble Classification Approach for Sarcasm Detection", Behavioural Neurology, Hindawi, Volume 2021, Pages 1-13
[12] Christopher Ifeanyi EkeID, Azah Anir Norman, Liyana Shuib, "Multi-feature fusion framework for sarcasm identification on twitter data: A machine learning based approach", PLOS ONE, 2021, Pages 1-32
[13] Yi Liu; Zengwei Zheng; Binbin Zhou; Jianhua Ma; Lin Sun; Ruichen Xia, "Multimodal Sarcasm Detection Based on Multimodal Sentiment Co-training", 2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles, 2023, Pages 508-515
[14] Amal Alqahtani, Lubna Alhenaki, Abeer Alsheddi, "Text-based Sarcasm Detection on Social Networks: A Systematic Review", (IJACSA) International Journal of Advanced Computer Science and Applications,Volume 14, No. 3, 2023, Pages 313-328
[15] Vandita Grover, Hema Banati, "An attention approach to emoji focused sarcasm detection", Heliyon, Elsevier, Volume 10, Issue 17, 2024, Pages 1-12
[16] Qusai Abuein, Ra'ed M. Al-Khatib, Aya Migdady, Mahmoud S. Jawarneh, Asef Al-Khateeb, "ArSa-Tweets: A novel Arabic sarcasm detection system based on deep learning model", Heliyon, Elsevier, Volume 10, Issue 17, 2024, Pages 1-14
[17] Rolandos Alexandros Potamias, Georgios Siolas, Andreas Georgios Stafylopatis, "A transformer-based approach to irony and sarcasm detection", Neural Computing and Applications, Springer, 2020, Volume 32, Pages 17309–17320
[18] Md Saifullah Razali, Alfian Abdul Halin 1, Lei Ye, Shyamala Doraisamy, And Noris Mohd Norowi, "Sarcasm Detection Using Deep Learning With Contextual Features", IEEE Access, 2021, Pages 68609-68618
[19] Derwin Suhartono, Alif Tri Handoyo and Franz Adeta Junior, "Feature-Based Augmentation in Sarcasm Detection Using Reverse Generative Adversarial Network", 2023, Computers, Materials & Continua 2023, Volume 77, Issue 3, Pages 3637-3657
[20] Akshi Kumar, Shubham Dikshit, and Victor Hugo C. Albuquerque, "Explainable Artificial Intelligence for Sarcasm Detection in Dialogues", Wireless Communications and Mobile Computing, Hindawi, Volume 2021, Pages 1-13 pages
[21] Muhammad Ehtisham Hassan, Masroor Hussain, Iffat Maab, Usman Habib, Muhammad Attique Khan, Anum Masood, "Detection of Sarcasm in Urdu Tweets Using Deep Learning and Transformer Based Hybrid Approaches", IEEE Access, Volume 12, 2024, Pages 1-14
[22] Derwin Suhartono, Wilson Wongso, Alif Tri Handoyo, "IdSarcasm: Benchmarking and Evaluating Language Models for Indonesian Sarcasm Detection", IEEE Access, Volume 12, 2024, Pages 87323 – 87332
[23] S. Muhammad Ahmed Hassan Shah, Syed Faizan Hussain Shah, Asad Ullah, Atif Rizwan, Ghada Atteia, Maali Alabdulhafith, "Arabic Sentiment Analysis and Sarcasm Detection Using Probabilistic Projections-Based Variational Switch Transformer", IEEE Access, Volume 11, 2023, Pages 67865 - 67881

[24] Avinash Kumar, Vishnu Teja Narapareddy, Veerubhotla Aditya Srikanth, Aruna Malapati, Lalita Bhanu Murthy Neti, “Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM”, IEEE Access, Volume 8, 2020, Pages 6388 – 6397
[25] Mochamad Alfan Rosid, Daniel Oranova Siahaan, Ahmad Saikhu, “Sarcasm Detection in Indonesian-English Code-Mixed Text Using Multihead Attention-Based Convolutional and Bi-Directional GRU”, IEEE Access, Volume 12, 2024, Pages 137063 – 137079
[26] New Approach to Underwater Image Enhancement Using Modified Residual Blocks in Generator Architecture for Improved Cycle Generative Adversarial Networks K Selvaraju, S Rajamani - Proceedings of the Bulgarian Academy of Sciences, 2024.