Paper

Multimodal Sarcasm Detection via Hybrid Classifier with Optimistic Logic

Dnyaneshwar Madhukar Bavkar[1], Ramgopal Kashyap[1], and Vaishali Khairnar[2]

[1]*Department of Computer Science and Engineering, Amity University, Raipur, Chhattisgarh, India,*
[2]*Department of Information Technology, Terna Engineering College, Nerul, Navi Mumbai, India.*

https://doi.org/10.26636/jtit.2022.161622

Abstract — This work aims to provide a novel multimodal sarcasm detection model that includes four stages: pre-processing, feature extraction, feature level fusion, and classification. The pre-processing uses multimodal data that includes text, video, and audio. Here, text is pre-processed using tokenization and stemming, video is pre-processed during the face detection phase, and audio is pre-processed using the filtering technique. During the feature extraction stage, such text features as TF-IDF, improved bag of visual words, n-gram, and emojis as well on the video features using improved SLBT, and constraint local model (CLM) are extraction. Similarly the audio features like MFCC, chroma, spectral features, and jitter are extracted. Then, the extracted features are transferred to the feature level fusion stage, wherein an improved multilevel canonical correlation analysis (CCA) fusion technique is performed. The classification is performed using a hybrid classifier (HC), e.g. bidirectional gated recurrent unit (Bi-GRU) and LSTM. The outcomes of Bi-GRU and LSTM are averaged to obtain an effective output. To make the detection results more accurate, the weight of LSTM will be optimally tuned by the proposed opposition learning-based aquila optimization (OLAO) model. The MUStARD dataset is a multimodal video corpus used for automated sarcasm discovery studies. Finally, the effectiveness of the proposed approach is proved based on various metrics.

Keywords — Bi-GRU, improved CCA, LSTM, multimodal sarcasm detection

Tab. 1. Nomenclature used.

Abbreviation	Description
AAM	Active appearance model
ALO	Ant lion optimization
AO	Aquila optimizer
BiGRU	Bi-directional gated recurrent unit
CAT	Convolution and attention
CCA	Canonical correlation analysis
CDVaN	Contextual dual-view attention network
CLM	Constraint local model
CMBO	Cat mouse-based optimization
CNN	Convolutional neural network
DL	Deep learning
DT	Decision tree
FDR	False discovery rate
FNR	False negative rate
FPR	False positive rate
HC	Hybrid classifier
IWAN	Incongruity-aware attention network
LBF	Local binary feature
LBP	Local binary pattern
LSTM	Long short term memory
MCC	Matthews's correlation coefficient
MFCC	Mel frequency cepstral coefficient
ML	Machine learning
NB	Naïve Bayes
NLP	Natural language processing
NN	Neural network
NPV	Net predictive value
OLAO	Opposition learning based aquila optimization
PCA	Principal component analysis
PRO	Poor and rich optimization
RF	Random forest
RNN	Recurrent neural network
SDS	Self-deprecating sarcasm
SLBT	Shape local binary texture
SSO	Social spider optimization
SVM	Support vector machine
TF-IDF	Term frequency-inverse document frequency

1. Introduction

Sarcasm is described as the use of remarks that imply the reverse of what one says, either to damage someone's feelings or to criticize something spectacularly [1]–[3]. It is a metaphorical language that is frequently used to communicate on social media, verbally and also with the use of the written text format. In the sarcasm sentiment, negative emotions are expressed via positive words found in the text, in order to expose their sarcasm [4], [5]. Tempo and speech time, variation, pitch level, and acoustic characteristics are all available in verbal sarcasm [6]. To demonstrate its sarcastic characteristics, this type of communication relies also

on tones and gestures, including eye and hand movements. Since no tune or gestures are available in sarcastic utterances represented in the text form, an ordinary person cannot recognize them. To detect sarcasm [7], an effective NLP approach is required for categorizing sarcastic features and properties within a sentence available in the text format [8]–[10].

Sarcasm was already characterized by NLP methods, where identification is described as the process of classifying a word or sentence sequence with sarcastic features and qualities by using NLP techniques [11]. It is also known as a system that learns and identifies ordinary and sarcastic sentences at the semantic level. Sentiment categorization is the basic goal of processes detecting sarcasm in a sentence. Due to its durability and ability to monitor itself based on specific datasets and requirements, the ML model [12]–[14] is frequently used for sarcasm detection [15], [16]. Sarcasm detection has proven to be useful in a variety of situations, as it allows businesses to analyze customers' reactions to their items, thus improving product quality [17]. It also aids in the elimination of incorrect categorization of customer views on problems, goods, and services. In human-computer interactions, sarcasm detection is also effective in conversation, system review rating, and summarization. For example, ML-based sarcasm identification [18] is used, relying on higher entropy, SVM, NN, window class, statistics, semantics, etc. In addition, an in-depth survey is conducted on automatic sarcasm detection methods, with a comparison of the scale of a given study, including the features, classification techniques, as well as performance parameters used. The survey is beneficial in identifying the newest trends in sarcasm detection [19]–[22].

The major contribution of this work is:

– BoW is newly defined along with other text-based features, like TF-IDF, n-gram,
– during the feature-level fusion phase, an improved multilevel CCA fusion technique is performed,
– OLAO model is implemented for weight optimization in LSTM.

In this work, a review of multimodal sarcasm detection methods is presented in Section 2. An overall description of the adopted multimodal sarcasm detection model is portrayed in Section 3. Pre-processing, feature extraction and level fusion processes are presented in Section 4. Section 5 describes a classification methods based on hybrid classifiers. Section 6 depicts the weight optimization of LSTM via an OLAO algorithm. The results are presented and discussed in Section 7. Section 8 concludes the paper, while Table 1 summarizes the nomenclature and abbreviations used.

2. Literature Review

Basavaraj *et al.* [23] suggested a method for detecting sarcasm in human words. The approach captures three types of data: voice, text, and temporal facial expressions to exploit the basic cognitive properties of human utterances. The data was unstructured because it contained dimensions of feelings and emotions that were used to produce sarcasm, with facial expressions being impacted by glottal and facial organs. The main effort focused on creating natural judgments in the prediction processes by employing cognitive data lineage information. It was difficult to identify sarcasm in genuine human conversations. Utilizing cloud resources, the multiclass NN model was applied as a soft cognition technique for detecting sarcasm. Voice cues and eye motions were examples of cognitive traits identified that might impact sarcasm detection.

Deepak *et al.* in [24] utilized DL in code-switch tweets to identify sarcasm, particularly in an Indian native language being a mixture of Hindi and English. The suggested system combined a softmax attention layer with Bi-LSTM and CNN for detecting real-time sarcasm. The SentiHindi feature vector was created employing pre-trained GloVe word embeddings and handmade features. The suggested softAttBiLSTM-feature-rich CNN model was compared and validated using performance assessment. With a classification accuracy of ~ 0.93 as well as an F-measure of ~ 0.89, the system from [24] surpasses baseline DL techniques.

Wu *et al.* [25] created IWAN – an approach which uses a scoring method to identify sarcasm by concentrating on word-level incongruity among modalities. This scoring process might give words with incongruent modality a higher weight. The approach could capture word-level incongruity, resulting in greater performance and interpretability. The authors have added word-level characteristics for detecting multimodal sarcasm. In the MUStARD dataset, they performed comprehensive comparison trials with 7 baseline models, but the model produced traditional outcomes. The benefits of the suggested IWAN algorithm were presented based on experimental findings that not only offered traditional performance on the MUStARD dataset but also provided interpretability benefits.

Kamal *et al.* [26] demonstrated a DL strategy for identifying SDS on Twitter. They suggested a new CAT-BiGRU framework that comprises input, embedding, convolutional, two attention layers, and BiGRU. The SDS-based semantic and syntactic features in the embedding layers are extracted by the convolutional layer. Amazon word embedding as well as affective space and two SenticNet-based computing resources were determined to test the effectiveness of the suggested system. The authors concluded that DL-based techniques can reliably detect SDS in social media content based on the experimental results.

Eke *et al.* [27] conducted an analysis of sarcasm identification and classification strategies based on performance standards, datasets, classification models, feature engineering, and pre-processing. Text articles were studied during the research, with an emphasis placed on context and content-based language elements. Accuracy and precision metrics of such classification techniques as SVM, NB, RF, maximum entropy, and DT algorithm were measured and evaluated.

Kumar *et al.* [28] analyzed an empirical investigation of DL and shallow methods for detecting sarcasm used in text datasets. Using three predictive learning models, over 20,000

postings from Reddit and Twitter from the benchmark SemEval 2015 Task 11 were identified as sarcastic or non-sarcastic in this study. To generate the output, the first framework was developed based on TF-IDF weighted, which was trained through three classifiers, including gradient boosting, multinomial NB, and RF, as well as ensemble voting. The investigation compared the three learning approaches to classifying sarcasm into two datasets. It was discovered that the Bi-LSTM scheme achieved the maximum score for Reddit and Twitter datasets.

Ren *et al.* [29] suggested a CDVaN sarcasm identification model based on the sarcasm creation process. They used CDVaN for capturing contextual semantic information as well as for making the distinction between positive and negative situations in sarcasm. In contrast to the sarcasm-generating process, a multi-hop attention network was used to acquire contextual semantic information. Investigations on IAC-V2 as well as IAC-V1 datasets have shown that the suggested CDVaN system was capable of efficiently discriminating sarcasm. The model achieved state-of-the-art or equivalent performance, as per the findings.

Zheng *et al.* [30] identified sarcasm and irony on Twitter using several NLP and ML approaches. They discussed several research projects concentrating on irony and sarcasm to evaluate and clarify the meanings of such terms. The experiment was carried out by comparing several types of classification algorithms relying on some well-known text classification classifiers. The findings of this experiment suggest that ML approaches, particularly DL methods, were on the rise as the most promising for classification-related tasks. The F-score of the result was 0.89 and is comparable to the F-score of the sarcastic dataset.

Table 2 summarizes research projects focusing on multimodal sarcasm detection. The NN model determined in [23] offers a lower mean error rate, a high accuracy level and higher sensitivity. However, experiments involving benchmark datasets were not conducted in this work. SoftArt BiLSTM-feature-rich CNN model from [24] offers a higher classification accuracy level, a better recall rate, higher precision, and higher F-scores, but this model could not overfit based on dropout regularization. Moreover, the IWAN model deployed in [25] offers better precision, a higher recall rate, the best

Tab. 2. Review of multimodal sarcasm detection systems.

Paper	Adopted scheme	Features	Limitations	Dataset used	Effectiveness values
[23]	NN model	Better accuracy, lower mean error, higher sensitivity	Experiments on benchmark datasets were not conducted	Multi-modal sarcasm detection dataset	Overall accuracy is 78.57%
[24]	SoftArt BiLSTM-feature-rich CNN method	Superior classification accuracy, higher recall, better precision, higher F-score	This model could not overfit based on dropout regularization	The randomly sampled dataset contains 3000 sarcastic and 3000 non-sarcastic bilingual Hinglish (Hindi English) tweets	Classification accuracy is 92.71%
[25]	IWAN model	Better precision, higher recall, best F1-score, improved interpretability	Context incongruity was not investigated	Multi-modal sarcasm detection dataset	Overall accuracy is 93%
[26]	CAT-BiGRU model	Higher precision, better recall, improved F-score, higher accuracy	Multilingual data operation was not performed on multimodal platforms	Six benchmark datasets including Twitter dataset	Overall accuracy is 90%
[27]	ML algorithm	Best classifier accuracy, increased precision, higher recall, maximum F-score	Lack of a standard dataset was an issue in sarcasm identification	Sarcasm identification dataset	F-score is 73.5%
[28]	Multinomial NB model	Highest accuracy, higher recall, better precision, increased F1-score	Crowd-sourced or self-tagging datasets provide novel limitations for detecting the sarcastic tone	SemEval 2015 Task 11 and Kaggle's Reddit dataset	Overall accuracy is 86.32%
[29]	CDVaN model	Good effectiveness, better performance, lower error rate	Sarcasm related work was not continued owing to multi-modal data	IAC-V1 dataset and IAC-V2 dataset	Precision level is 76.32%
[30]	CNN model	Higher F-score, higher accuracy, larger correct rate	Different pre-processing approaches were not explored based on irony as well as sarcasm recognition	Semantic evaluation 2018 task 3: irony detection in English tweets	F1-score is 0.99%

F1-score, and improved interpretability. However, it failed to investigate context incongruity. Likewise, the CAT-BiGRU model from [26] offers higher precision, a better recall rate, an improved F-score, and higher accuracy. However, no multilingual data operations were performed on multimodal platforms. The ML algorithm was exploited in [27] and it has been determined that it offers the best classifier accuracy, an increased precision level, a higher recall rate and a maximum F-score. However, the lack of a standard dataset was an issue in sarcasm identification. The multinomial NB model from [28] offers the highest accuracy level. However, crowd-sourced or self-tagging datasets provide novel limitations related to detecting the sarcastic tone. The CDVaN model proposed in [29] is characterized by a lower error rate and ensures better performance and effectiveness. However, the sarcasm work was not continued due to multimodal data. Finally, the CNN model presented in [30] ensures better results, but different pre-processing approaches were needed to assure the quality of input data.

3. Multimodal Sarcasm Detection Model Adopted

This work introduces a new multimodal sarcasm detection model that comprises pre-processing, feature extraction, feature level fusion, and classification stages. First, the input text, video, and audio are subjected to the pre-processing stage. Next, the text content is pre-processed using tokenization and stemming. Video is pre-processed via face detection (Viola-Jones), and audio is pre-processed using the filtering technique (Butterworth filtering). Subsequently, the pre-processed text, video, and audio inputs are transferred to the feature extraction stage, where text features are extracted using TF-IDF, improved bag of words, n-gram, and emojis. Video features are extracted via improved SLBT and CLM. Audio features are extracted using MFCC, chroma, spectral features, and jitter. The extracted features are transferred to the feature level fusion phase, wherein an improved fusion technique is adopted. Classification is performed using a hybrid classifier

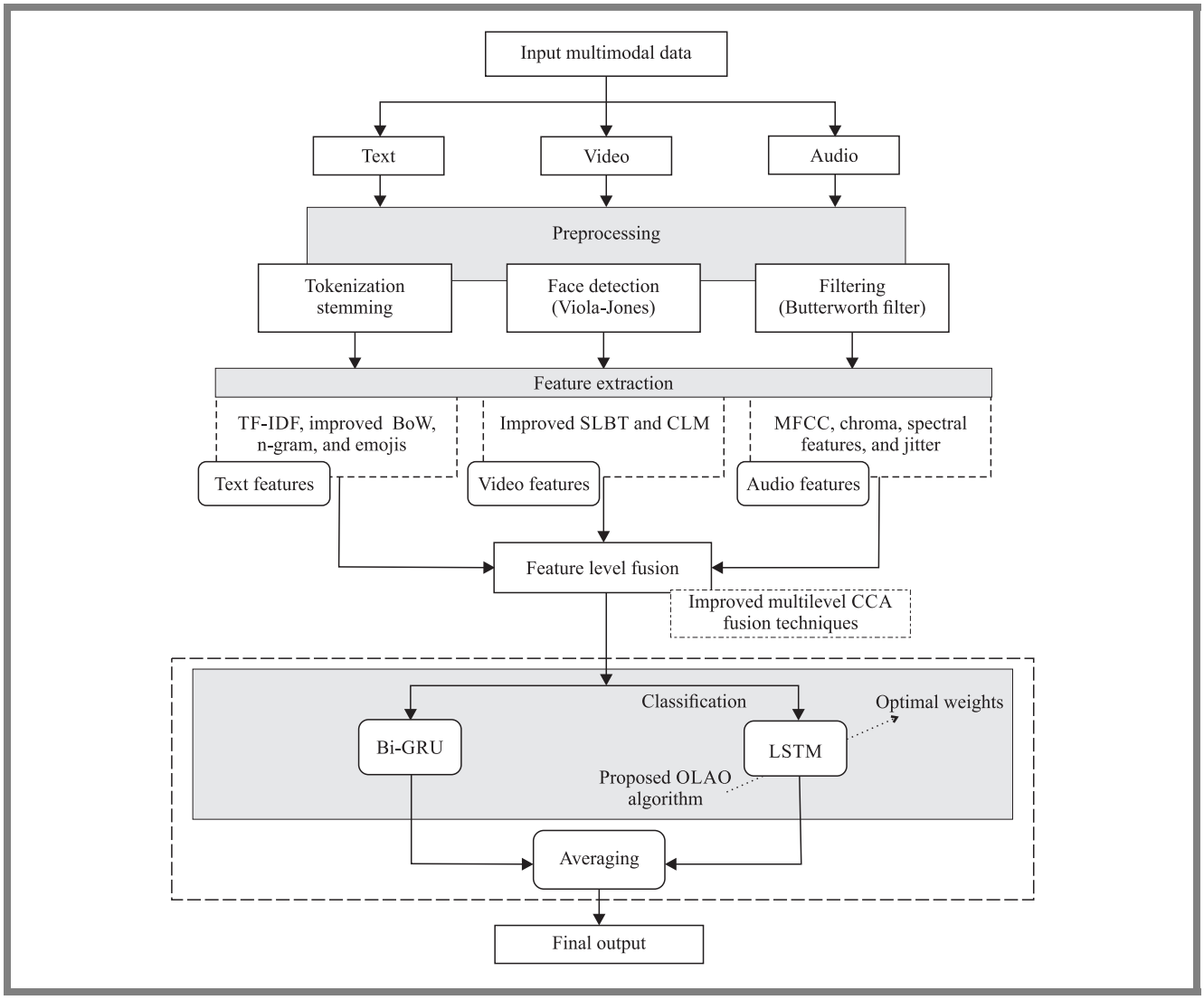

Fig. 1. Overall framework of the adopted model.

that combines LSTM and Bi-GRU by averaging the output of LSTM and Bi-GRU. To make the detection more precise and accurate, the weights of LSTM are tuned by a self-improved AO algorithm. The results show the presence of sarcasm in the given input.

Figure 1 illustrates the overall architecture of the adopted multimodal sarcasm detection model.

4. Model Details

4.1. Pre-processing Stage

Pre-processing is the initial and crucial process for successful learning. The input data is the multimodal data that includes text, audio, and video. Text is pre-processed using tokenization and stemming. Video is pre-processed via the face detection (Viola-Jones) model, and audio is pre-processed using the filtering technique (Butterworth filtering).

Tokenization [31] is the method of transforming text into tokens prior to vectorization. Undesirable tokens may be easily filtered off. For instance, a document may be divided into paragraphs or phrases broken down into words. This method consists in dividing large amounts of text into smaller chunks. Raw texts are broken down into words and phrases during the tokenization process as well. As a consequence, the tokens might aid in determining the NLP framework or understanding the context. By analyzing the word sequence, tokenization aids in determining the meaning of the text.

Tokenization may be accomplished using a variety of libraries and approaches. This task is carried out using such libraries as Keras, NLTK, and Gensim.

Stemming [31] is one of the normalizing strategies that reduce the number of calculations. It is a strategy for removing suffixes and retrieving the original words. During the stemming procedure, libraries such as Snowball Stemmer, Porter Stemming, and others are employed. Furthermore, stemming is mostly used to reduce data dimensionality. Stemming-related errors include under- and over-stemming.

Under stemming is characterized as false negatives which occur if 2 words are stemmed from the same stems and their roots are similar. Over stemming is viewed as a false-positive case that occurs when two words are stemmed from the same root but have different stems. Stemming is relied upon up information retrieval systems (i.e. Internet search engines) and other applications. It is also used in domain analysis to identify the existing domain vocabularies.

Face detection is difficult due to the numerous differences in the appearance of individual images, including facial expressions, pose variations, image orientation, occlusion, as well as lighting conditions. The Viola-Jones face detection method [32] is employed in this study.

It is an object detection approach capable of operating in real-time. Full view frontal upright faces are required for Viola-Jones. The approach, at its most basic level, reads an input image via a window, seeking human facial characteristics. When more characteristics are detected, the window in an image is classified as a face. Further, the window should be resized and the process should be repeated to produce different size faces. For each window scale, the procedure is applied separately from other scales. To reduce the number of features, each window must be checked using a series of levels. Earlier levels have fewer features to verify and are thus simpler to pass, whereas later levels have more features and therefore are more difficult. The examination of features is performed at each level, and if the collected value does not meet the threshold, the level is considered failed and the specific window is not recognized as a face. The Viola-Jones face detection approach is divided into three key stages (integral image, classifier learning with AdaBoost, and attentional cascade structure) that allow for successful face detection in real-time applications.

The Butterworth filter [33] has a frequency response in the pass band that would be as flat as feasible. A maximally flat magnitude filter is another name of this particular filter. The Butterworth family of filters is very simple and useful. The cutoff frequency and filter order are the two key parameters used. Frequency response is monotonic and filter order affects the sharpness of the transition from the pass band to the stop band.

The poles linked with the squares of the frequency response magnitude are uniformly distributed in angle on concentric with the origin circle in the s-plane and containing a radius equal to the cut-off frequency for continuous time Butterworth filter. The poles that characterize the system function are easily acquired once the cutoff frequency and the filter order are established. One may easily design a differential equation that characterizes the filter after the poles have been determined. The squared magnitude function for an m-th order Butterworth low pass filter is:

$$|C(\mathrm{j}\omega)|^2 = C(\mathrm{j}\omega) \times C^*(\mathrm{j}\omega) = \frac{1}{1 + \left(\frac{\mathrm{j}\omega}{\mathrm{j}\omega_c}\right)^{2m}}. \tag{1}$$

The first $2m-1$ derivatives of $C(\mathrm{j}\omega)^2$ at $\omega = 0$ are equal to 0 and the Butterworth response is maximally flat at $\omega = 0$. The derivative of the magnitude response is always negative for $+\omega$ and the magnitude response is minimized with ω. For $\omega >> \omega_c$ the magnitude response is determined by:

$$|C(\mathrm{j}\omega)|^2 = \frac{1}{\left(\frac{\mathrm{j}\omega}{\mathrm{j}\omega_c}\right)^2}. \tag{2}$$

4.2. Feature Extraction

The pre-processed text, video, and audio obtained are subjected to the feature extraction phase. From the text, such features as TF-IDF, n-grams, improved BoW, and emojis are extracted. TF-IDF [34] is a significant text demonstration format and includes a longer history when compared with the 3 well-known depiction techniques. It depends upon the BOW method, where a text is characterized by a compilation of words deployed in the document. The TF_{pq} constraint describes how many times word p appears in the document q. The better the value, the more noteworthy the word. The DFp constraint signifies the count of documents where p appears

once. If p is significant for q, it must comprise a higher TF_{pq} and lower DF_p. Hence, TF-IDF is determined as:

$$\text{TD-IDF}_{pq} = \mathrm{TF}_{pq} \log \frac{M}{\mathrm{DF}_p + 1}. \tag{3}$$

The extracted TF-IDF features are denoted as TF-IDF.

Any sequence of n tokens or words is called an n-gram. Moreover, an n-gram model [44] is defined as "a method of including sequences of words or characters that permits us to maintain richer pattern discovery in text, i.e. it attempts to captivate patterns of sequences (words or characters subsequent to one another) while being responsive to appropriate relations (words or characters subsequent to one another)". The extracted n-gram-based features are denoted as *Ngram*.

BoW is the simplest technique used to transform the text into features. It separates words in the reviewed text into word count data and calculates the number of times a phrase appears in the corpus of a given text. It only cares about the order in which words appear in the text, not the sequence in which they appear frequently. The existing BoW evaluation does not consider the semantics of visual words, which is considered in the improved evaluation.

In the improved BoW, histograms of the visual words are used as a feature vector, such as:

$$K\left(P, Q\right) = \sum_{l=1}^{L} k(P_l, Q_l), \tag{4}$$

Where P and Q are images and l is visual word number. Then:

$$K(P_l, Q_L) = J_l^I + \sum_{i=0}^{I-1} \frac{1}{2^{I-l}} \left(J_l^i - j_l^{i+1}\right) S, \tag{5}$$

where I denotes the count and i indicates the current levels. Each level is weighted using $\frac{1}{2^{I-l}}$, J_l^i indicates the histogram intersection function, and S is the scaling factor.

$$J\left(H_{P_l}^i, H_{Q_L}^i\right) = \sum_{k=1}^{4^i} \min\left(H_{P_l}^i(k), H_{Q_l}^i(k)\right), \tag{6}$$

where the $H_{P_l}^i(k)$ denotes l-th count of visual words in the k-th subregion of image P at level i. The improved BoW characteristics are denoted by the *IBoW* symbol.

Similarly, for texts with emojis a sentiment score based on unicode is extracted together with the text features, with regard to the emoji's lexicon.

The position of an emoji is determined by its sentiment score as well as neutrality. The emotion score is between -1 and $+1$. Positive emojis are on the right-hand side of the map, while negative emojis are on the left-hand side. The most prevalent negative emoji is a sad face. The most common positive emojis include trophies, celebration symbols, hearts and a wrapped present – in addition to joyful smiles. Neutral emojis are classified using the neutrality range of 0 to 1 and all emojis have a sentiment score of 0. The extracted text with emoji features is denoted as EMO.

The overall extracted text features are denoted as TF $=$ TF-IDF $+$ *Ngram* $+$ *IBoW* $+$ EMO.

4.2.1. Video-based Features

From video content, features like improved SLBT and CLM are extracted. SLBT [10] is a feature that merges texture and shape characteristics. SLBT is identical to AAM, because it analyzes texture modeling using LBP texture features rather than intensity values. Consider $\mathrm{IM} = [\mathrm{IM}_1, \mathrm{IM}_2, \ldots, \mathrm{IM}_{\mathrm{NO}}]$ symbolizing a training set of pictures NO with $\mathrm{XP} = [\mathrm{XP}_1, \mathrm{XP}_2, \ldots, \mathrm{XP}_{\mathrm{NO}}]$ as shape landmark points. By matching these landmark points and then performing PCA on those points, shape variants may be achieved. Equation (7) determines any shape vector XP in the training set:

$$\begin{aligned} \mathrm{XP} &\approx \overline{\mathrm{XP}} + \mathrm{RS}_{ls}\mathrm{BD}_{ls}, \\ \mathrm{BD}_{ls} &= \mathrm{RS}_{ls}^{\hat{T}} \left(\mathrm{XP} - \overline{\mathrm{XP}}\right), \end{aligned} \tag{7}$$

where $\overline{\mathrm{XP}}$ denotes the mean shape, RS_{ls} includes the eigenvectors of the largest eigenvalues Ω_{ls} and BD_{ls} denotes weights or shape model parameters (e.g. ls denotes the shape in BD_{ls}). Such an approach may capture shape model parameters matching a given image by modifying Eq. (7).

To generate a shape-free patch, each training set image is warped into a mean shape for texture modeling. Computational complexity, efficiency, as well as processing time are mostly influenced by the size of the shape-free patch. For texture modeling in AAM, direct intensity values from a shape-free patch are required. To acquire illumination and noise invariant features, SLBT conducts LBP on a shape-free patch. Feature extraction using LBP is easier and faster than with Gabor wavelets.

Moreover, the shape vector and LBP vectors are used in SLBT. Unlike the LBP evaluation used in the conventional technique, improved LBP (geometric mean-LBP) is based on the comparison with neighboring pixels after comparison of the regional average *RM* of the image with the center pixel. Here, G_g indicates the neighboring pixel, s indicates the number of neighbors. The operation logic of ILBP is:

$$\mathrm{ILBP} = \sum_{g=1}^{\delta} t\, 2^{g-1}. \tag{8}$$

In improved LBP, function t is determined by:

$$t = \begin{cases} 1, & \text{if } GM \geqslant G_b \text{ and } RM \leqslant G_g \\ 0, & \text{else if } RM \geqslant G_b \text{ and } RM > G_g \\ 1, & \text{else if } RM < G_b \text{ and } G_b \leqslant G_g \\ 0, & \text{otherwise} \end{cases}, \tag{9}$$

$$GM = \left(\prod_{g=1}^{s} G_g\right)^{\frac{1}{s}}, \tag{10}$$

where G_b indicates the center pixel and G_g denotes the neighboring pixel. The improved SLBT characteristics are denoted by the ISLBT symbol.

Consider $\mathrm{LI} = [\mathrm{LI}_1, \mathrm{LI}_2, \ldots, \mathrm{LI}_{\mathrm{NO}}]$ as the LBP feature histogram of all training sample images. Texture modeling (same as shape modeling) is accomplished with PCA given in Eq. (11). Here, $\mathrm{BD}_{\hat{t}}$ denotes the weights or texture mod-

eling parameter ($\breve{t}$ refers to texture in $\mathrm{BD}_{\breve{t}}$), $\mathrm{RS}_{\breve{t}}$ indicates the eigenvectors and LI refers to the mean vector.

$$\mathrm{BD}_t = \mathrm{RS}_t^T \left(\mathrm{LI} - \overline{\mathrm{LI}}\right). \tag{11}$$

Using Eq. (12), a mixed shape and texture parameter vector are generated. Because shape (distance) and texture (intensity values) are measured using separate units, a diagonal matrix of weights WE_{ls} is computed for each shape parameter. By using PCA on the combined parameter vector as in Eq. (13), the shape texture parameter determining the texture, and local shape may be achieved.

$$\mathrm{BD}_{lst} = \begin{pmatrix} \mathrm{WE}_{ls} & \mathrm{BD}_{ls} \\ \mathrm{BD}_t & \end{pmatrix}, \tag{12}$$

$$\mathrm{CZ} = \mathrm{RS}_{lst}^T \left(\mathrm{BD}_{lst} - \overline{\mathrm{BD}_{lst}}\right). \tag{13}$$

In Eqs. (12), (13) RS_{lst} denotes the eigenvectors, $\overline{\mathrm{BD}_{lst}}$ specifies the mean vector and CZ refers to the shape texture parameter. The LBF feature histogram is derived from a shapeless patch with five divisions along each row or column (e.g. 25 blocks).

If an annotated test image XP_{test} is provided as input, Eq. (7) is used to convert it into the shape model parameter BD_{test} which is then multiplied by WE_{ls}. The test image is distorted into a shapeless patch, by which LBP features are extracted as LI_{test} and the texture model parameter BD_{test} is computed by Eq. (11). From Eq. (13) CZ_{test} is employed for classification purposes and is generated by combining $\mathrm{BD}_{ls\,test}$ and $\mathrm{BD}_{t\,test}$ results in $\mathrm{BD}_{lst\,test}$ as well as the shape texture parameter CZ_{test}.

CLM [36] is a group of approaches for identifying collections of points on a target picture that are bound by a statistical shape model. The main procedure aims to:

– sample a section of the image surrounding the current estimate and project it into a reference frame,
– create a "response image" for each point that shows the cost of having the point at each pixel,
– use the shape model parameters to identify a combination of points that minimizes the cost.

The optimum fit is discovered by minimizing the shape and pose parameters:

$$B(a,d) = \sum_{r=1}^{r=e} R_r \left[\hat{T}_d(X_r + Y_r a)\right]. \tag{14}$$

The term CLM is mainly referred to as a model used for creating response images using normalized correlation with a local patch, with the model patches being updated to match the current face while simultaneously being constrained by a global texture model. The CLM features are denoted by CLM.

The overall extracted video features are denoted as VF = ISLBT + CLM.

4.2.2. Audio-based Features

Features such as MFCC, chroma, spectral features, and jitter are extracted from audio content.

MFCCs [37] are frequently employed in speech recognition systems that can automatically recognize digits spoken into a phone. MFCCs are rapidly being used in music-related information retrieval applications, such as genre categorization apps and audio similarity measurements. The way you use the oral anatomy to produce each sound determines how it sounds. As a consequence, creating a description that encapsulates the physical mechanics of spoken language is one technique capable of uniquely identifying sounds. The method of encoding this data is to use MFCC features. The basic MFCC properties of the signal are provided by cepstral coefficients. On the other hand, additional characteristics, such as delta, acceleration, and energy can typically increase the accuracy. MFCC-based audio features are denoted by the MFCC symbol.

The 12 various pitch classes are referred to as chroma [38] features or chromagrams. Chroma-based characteristics, referred to as "pitch class profiles", are useful for evaluating music with usefully classified pitches (typically in 12 groups) and for tuning which approximates the equal-tempered scale. Chromatic and melodic features of music are captured by chroma features which are responsive to changes in timbre as well as accompaniment. Chroma audio-based features are denoted by the CHR symbol.

Frequency and power characteristics of the signal are extracted using the spectral features block [39]. Filters may be used to remove undesirable frequencies. Such an approach is ideal for analyzing repeating signal patterns, including motions or vibrations from an accelerometer. Spectral characteristics are denoted by the SP symbol.

Jitter defines time distortions of phase and amplitude of the signal caused by clock deviation introduced during the analog-to-digital conversion. The effect of jitter increases with transmission distance and with the number of signal conversion stages. Jitter features are represented as Jitter.

The extracted audio features are denoted as AF = MFCC + CHR + SRP + *Jitter*. All extracted features combined are denoted by the FE symbol:

$$\mathrm{FE} = \mathrm{TF} + \mathrm{VF} + \mathrm{AF}. \tag{15}$$

4.3. Feature Level Fusion

The extracted text, video, and audio features are subjected to the feature-level fusion process. First, the audio and video features are transferred to CCA1 that produces an output and then the text features are transferred to CCA2. Next, the combined outcome of CCA1 and CCA2 is handed over to CCA3 to produce the final feature level fusion output. Figure 2 illustrates the feature-level fusion process.

Multilevel CCA [40] is a technique used for performing multivariate statistical analysis. The goal of CCA is to project two groups of multivariate data into an ordinary space with the highest possible correlation among them. The purpose of CCA in this situation is to discover a couple of column projection vectors $u_V \in \Re^{\hat{d}}$ and $u_Z \in \Re^{\hat{d}}$ in which the correlation among $u_V^T V$ and $u_Z^T Z$ is maximized, given

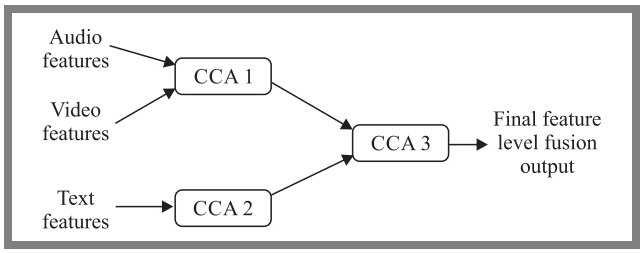

Fig. 2. Feature-level fusion process using multilevel CCA.

by two data matrices $V = \{V_v \in \Re^{\hat{d}}, v = 1, 2, \ldots, \hat{K}\}$, $Z = \{Z_v \in \Re^{\hat{d}}, v = 1, 2, \ldots, \hat{K}\}$, and $\hat{K}$ pairings of data from two modalities. The objective function defined as the maximization function in this case is:

$$\arg\max_{u_{\hat{L}}, u_{\hat{S}}} \frac{u_V^T \hat{C}_{\text{ZZ}} u_z}{\sqrt{u_V^T \hat{C}_{\text{VV}} u_V} \sqrt{u_Z^T \hat{C}_{\text{ZZ}} u_Z}}. \tag{16}$$

The data covariance matrices are $\hat{C}_{\text{VV}} = E[\text{VV}]^T$, $\hat{C}_{\text{ZZ}} = E[\text{ZZ}]^T$, and $\hat{C}_{\text{VZ}} = E[\text{VZ}]^T$.

$$\begin{aligned} &\text{maximize } u_V^T \hat{C}_{\text{VZ}} u_Z \\ &\text{Subject to } u_V^T \hat{C}_{\text{VV}} u_V = 1 \cdot \\ &\qquad\qquad u_Z^T \hat{C}_{\text{ZZ}} u_Z = 1 \end{aligned} \tag{17}$$

Equation (17) is solved via generalized eigenvalue issues:

$$\begin{bmatrix} 0 & \hat{C}_{\text{VZ}} \\ \hat{C}_{ZV} & 0 \end{bmatrix} \begin{bmatrix} u_V \\ u_Z \end{bmatrix} = \lambda \begin{bmatrix} \hat{C}_{\text{VV}} & 0 \\ 0 & \hat{C}_{\text{ZZ}} \end{bmatrix} \begin{bmatrix} u_V \\ u_Z \end{bmatrix}, \tag{18}$$

where u_V denotes an eigenvector of $\hat{C}_{\text{VV}}^{-1} \hat{C}_{\text{VZ}} \hat{C}_{\text{ZZ}}^{-1} \hat{C}_{ZV}$ and u_Z indicates an eigenvector of $\hat{C}_{\text{ZZ}}^{-1} \hat{C}_{ZV} \hat{C}_{\text{VV}}^{-1} \hat{C}_{\text{VZ}}$. The projection matrices U_V and U_Z are attained via stacking u_V and u_Z as column vectors, respectively to various eigenvalue issues.

An improved correlation is determined in multi-level CCA as:

$$Icorr = 1 - \left(\frac{\sum_{\bar{c}=1}^{\bar{Q}} \left(\tilde{P}_{\bar{c}} - \bar{P}\right)\left(\tilde{R}_{\bar{c}} - \bar{R}\right)}{\sqrt{\sum_{\bar{c}=1}^{\bar{Q}} \left(\tilde{P}_{\bar{c}} - \bar{P}\right)^2} \sqrt{\sum_{\bar{c}=1}^{\bar{Q}} \left(\tilde{R}_{\bar{c}} - \bar{R}\right)^2}} \right)^2. \tag{19}$$

5. Classification via Hybrid Classifiers

After the feature-level fusion, the classification process is performed using an optimized hybrid classifier (Bi-GRU and LSTM). Then, the outputs of both classifiers are averaged to determine the final outcome.

Through the use of a linear connection and a gate control unit, the LSTM network offers an efficient way to solve gradient desertion-related difficulties. As a consequence, the LSTM network caught the time-series data's significant dependence. The sequences of persistent LSTM cells are included in LSTM [41] development. The input, output, and forget gates were all represented by three units in the LSTM cells. This feature enables the LSTM memory cells to suggest and store information for long periods of time.

Consider $\tilde{H}$ and $\tilde{C}$ as the hidden and cell state. Then $\tilde{H}_{\hat{t}}, \tilde{C}_{\hat{t}}$ and $F_{\hat{t}}, \tilde{C}_{\hat{t}-1}, \tilde{H}_{\hat{t}-1}$ represent the output and input layers, respectively. At time $\hat{t}$ the output, input and forget gates are denoted as $O_{\hat{t}}, \hat{I}_{\hat{t}}, \hat{G}_{\hat{t}}$ respectively. The LSTM cell is primarily used $\hat{G}_{\hat{t}}$ to filter the data. The modeling of $\hat{G}_{\hat{t}}$ is:

$$\hat{G}_t = \kappa\left(W_{\hat{L}} F_{\hat{t}} + h_{\hat{L}} + W_{\hat{J}} \tilde{H}_{\hat{t}-1} + h_{\hat{J}}\right), \tag{20}$$

where $W_{\hat{J}}, h_{\hat{J}}$ and $W_{\hat{L}}, h_{\hat{L}}$ specify the bias parameters and the weight matrix, respectively. The bias parameter and the weight factor are chosen randomly, while the weight factor is tuned optimally by he proposed OLAO model. The activation function of gate κ is elected as a sigmoid operation. Next, the LSTM cell makes use of the input gate to combine the proper data, as determined in Eqs. (21)-(23). $W_{\hat{X}}, h_{\hat{X}}$ and $W_{\hat{Y}}, h_{\hat{Y}}$ denote the weight matrices and the bias parameters which map the input and the hidden layers to the cell gate. W_x, h_x and W_y, h_y represent the weight and bias parameters that map the hidden and input layers to $\text{IL}_{\hat{t}}$:

$$\tilde{U}_{\hat{t}} = \tanh\left(W_{\hat{Y}} F_{\hat{t}} + h_{\hat{Y}} + W_{\hat{X}} \tilde{H}_{\hat{t}-1} + h_{\hat{X}}\right), \tag{21}$$

$$\text{IL}_{\hat{t}} = \kappa\left(W_y F_{\hat{t}} + h_y + W_x \tilde{H}_{\hat{t}-1} + h_x\right), \tag{22}$$

$$\tilde{C}_{\hat{t}} = \hat{G}_{\hat{t}} \tilde{C}_{\hat{t}-1} + \text{IL}_{\hat{t}} \hat{f}_{\hat{t}}, \tag{23}$$

Finally, the LSTM obtains a hidden layer (output) from the output gate as:

$$o_{\hat{t}} = \kappa\left(W_{\hat{e}} F_{\hat{t}} + h_{\hat{e}} + W_{\hat{r}} \tilde{H}_{\hat{t}-1} + h_{\hat{r}}\right), \tag{24}$$

$$\tilde{H}_{\hat{t}} = o_{\hat{t}} \tanh\left(\tilde{C}_{\hat{t}}\right), \tag{25}$$

where $W_{\hat{e}}, h_{\hat{e}}$ and $W_{\hat{r}}, h_{\hat{r}}$ indicate the weight and bias parameters used for mapping the input and hidden layers to $o_{\hat{t}}$ respectively. The output of LSTM is denoted as CL_{LSTM}.

For organizing the sequential data stream, learning a continuous representation might be beneficial. An RNN is dedicated to encoding sequential data. Here, a Bi-GRU for learning the features from a sentence sequence, with GCN appending the outputs for DDI extraction afterward, is used. Bi-GRU [42] is broken down into two sections for calculation: forward and reverse sequence information transfers. The forward GRU for a given sentence $\tilde{Z} = (z_1, z_2, \ldots, z_n)$, $z \in \Im^{\tilde{k}}$, z signifies the current word concatenating vector. The forward GRU is:

$$\hat{i} = \sigma\left(w_{\tilde{y}\hat{i}} \tilde{y}_{\tilde{g}} + w_{\hat{h}\hat{i}} \hat{h}_{\tilde{g}-1} + \tilde{a}_{\hat{i}}\right), \tag{26}$$

$$\tilde{l} = \sigma\left(w_{\tilde{y}\tilde{l}} \tilde{y}_{\tilde{g}} + w_{\hat{h}\tilde{l}} \hat{h}_{\tilde{g}-1} + \tilde{a}_{\tilde{l}}\right), \tag{27}$$

$$\tilde{s} = \tanh\left(w_{\tilde{y}\tilde{s}} \tilde{y}_{\tilde{g}} + w_{\hat{h}\tilde{s}} \left(\hat{i}\Theta\right) \hat{h}_{\tilde{g}-1} + \tilde{a}_{\tilde{s}}\right), \tag{28}$$

$$\hat{h} = \left(1 - \tilde{l}\right) \Theta \hat{h}_{\tilde{g}-1} + \tilde{l} \Theta \tilde{s}, \tag{29}$$

where w_* and $\tilde{a}_*$ are the weight matrix and the bias vector, respectively, σ denotes the sigmoid function, $\hat{h}_{\tilde{g}}$ is the hidden state of the current time step $\tilde{g}$, Θ is element-wise multiplication, and $\tilde{y}_{\tilde{g}}$ is the input word vector at time step $\tilde{g}$, $\vec{h}_{\hat{i}}$ and $\overleftarrow{h}_{\hat{i}}$ indicate the forward GRU and backward GRU output, respectively. The Bi-GRU output is indicated as $\hat{h}_{\hat{i}}^{\text{Bi-GRU}} = \left[\vec{h}_{\hat{i}}; \overleftarrow{h}_{\hat{i}}\right]$. The final classification output is:

$$\text{Out} = \frac{\text{CL}_{\text{LSTM}} + \hat{h}_{\hat{i}}^{\text{Bi-GRU}}}{2}. \tag{30}$$

6. LSTM Weight Optimization via OLAO

The weights of LSTM are tuned to optimal levels via the OLAO method adopted. Figure 3 illustrates the input solution to the adopted OLAO model. The total count of weights in LSTM is indicated as N. The final outputs of both Bi-GRU and LSTM are averaged to obtain the overall outcome. The error function is determined as error $= (1 - \text{accuracy})$. The objective function *Obj* of the implemented scheme is:

$$Obj = \min(\text{error}). \tag{31}$$

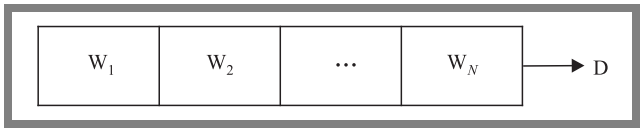

Fig. 3. Solution encoding.

6.1. Proposed OLAO Algorithm

Despite AO [43] offering better exploration capabilities, a good chance of reaching the optimal solution, and good exploitation-related abilities, it suffers from insufficient local exploitation ability. To overcome this problem, the OLAO model is proposed as an enhancement of the existing optimization models [44], [45]–[48]. AO is inspired by the behavior of hunting Aquila birds. The proposed OLAO concept deploys an OBL solution [49] that is modeled for generating opposite solutions. Specific points and their opposites are calculated simultaneously to select the best solution. OBL-based initialization guarantees an improved convergence rate, thus quickly reaching enhanced solutions.

6.2. Initialization of Solutions

The optimization rule relied upon in AO is a population-based method that starts with a population of candidate solutions D, as shown in Eq. (32). The said population is created stochastically between the lower *LB* and upper *UB* bounds of the specific issue. In each iteration, the best answer obtained is selected as the roughly optimal solution.

$$D = \begin{bmatrix} \tilde{q}_{1,1} & \cdots & \tilde{q}_{1,j} & \tilde{q}_{1,\text{Dim}-1} & \tilde{q}_{1,\text{Dim}} \\ \tilde{q}_{2,1} & \cdots & \tilde{q}_{2,\tilde{\jmath}} & \cdots & \tilde{q}_{2,\text{Dim}} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{q}_{A-1,1} & \cdots & \tilde{q}_{A-1,\tilde{\jmath}} & \cdots & \tilde{q}_{A-1,\text{Dim}} \\ \tilde{q}_{A} & \cdots & \tilde{q}_{A,\tilde{\jmath}} & \tilde{q}_{A,\text{Dim}-1} & \tilde{q}_{A,\text{Dim}} \end{bmatrix}, \tag{32}$$

where D indicates the group of current candidate solutions that are created randomly in Eq. (33), $D_{\tilde{\imath}}$ represents the position of the $\tilde{\imath}$-th solution, A denotes the entire count of candidate solutions, and Dim refers to the dimension of the issue.

$$\begin{aligned} D_{\tilde{\imath}\tilde{\jmath}} = \text{rand} \times (UB_{\tilde{\jmath}} - LB_{\tilde{\jmath}}) + LB_{\tilde{\jmath}}, \\ \tilde{\imath} = 1, 2, \ldots, A, \;\; \tilde{\jmath} = 1, 2, \ldots, \text{Dim}, \end{aligned} \tag{33}$$

where rand denotes a random number, $LB_{\tilde{\jmath}}$ indicates the $\tilde{\jmath}$-th lower bound, and $UB_{\tilde{\jmath}}$ refers to the $\tilde{\jmath}$-th upper bound of the issue.

6.3. AO mathematical Model

The proposed AO method imitates the behavior of Aquila's during each stage of the hunting at process. If $\tilde{l} \leqslant \frac{2}{3}\tilde{L}$ the exploration phases are exciting, it could move from exploration to exploitation steps using different behaviors, else, the exploitation phases are done.

The behavior of Aquilas is represented as a mathematical optimization framework whose goal is to find the optimum solution taking into consideration a given set of constraints. The mathematical model of the AO algorithm is determined as follows.

Step 1. Expanded exploration D_1. In the first model D_1, the Aquila identifies the its and chooses the optimal hunting location by soaring high in a vertical stoop. The AO requires high explorers to determine the area of the search space in which the prey is located. This behavior is represented as:

$$\begin{aligned} D_1(\tilde{l}+1) \;=\; & D_{\text{best}}(\tilde{l}) \cdot \left(1 - \frac{\tilde{l}}{\tilde{L}}\right) \\ & + \left[D_{\tilde{M}}(\tilde{l}) - D_{\text{best}}(\tilde{l}) \cdot \text{rand}\right], \end{aligned} \tag{34}$$

where $D_1(\tilde{l}+1)$ denotes the next iteration of the t solution that is produced by the initial search technique D_1. This indicates the approximate location of the prey and $D_{\text{best}}(\tilde{l})$ is the best solution obtain until the $\tilde{l}$-th iteration. Expression $\frac{1-\tilde{l}}{\tilde{L}}$ is often used to regulate the number of iterations in the extended search (exploration). $D_{\tilde{M}}(\tilde{l})$ indicates the mean value of the current solutions linked at the time of iteration $\tilde{l}$-th, as given by Eq. (35). $\tilde{l}$ and $\tilde{L}$ represent the current iteration as well as the higher number of iterations, respectively.

$$D_{\tilde{M}}(l) = \frac{1}{A}\sum_{\tilde{\imath}=1}^{A} D_{\tilde{\imath}}(\tilde{l}) \quad \forall \tilde{\jmath} = 1, 2, \ldots, \text{Dim}, \tag{35}$$

where Dim denotes the dimension of the issue and A indicates the count of candidate solutions (population size).

Step 2. Narrowed exploration D_2. Whenever the prey location is determined by a higher soar, the Aquila circles around the target, surveys the land and attacks using the second method D_2. In anticipation of the attack, AO carefully investigates the specific region of the targeted prey. This behavior is formulated as:

$$\begin{aligned} D_2(\tilde{l}+1) \;=\; & D_{\text{best}}(\tilde{l}) \times \text{Levy}(\beta) \\ & + D_{\tilde{R}}(\tilde{l}) + (\vec{u} - \vec{v}) \cdot \text{rand}, \end{aligned} \tag{36}$$

where $D_2(\tilde{l}+1)$ denotes the next iteration of $\tilde{l}$ solution, as determined by the second search procedure D_2. β indicates the dimension space, the Levy flight distribution $\text{Levy}(\beta)$ functions given by Eq. (37), while $D_{\tilde{R}}(\tilde{l})$ denotes a random solution from the $1, \ldots, A$ at $\tilde{l}$-th iteration.

$$\text{Levy}(\beta) = \bar{s} \cdot \frac{\bar{h} \times \vartheta}{|\bar{g}|^{\frac{1}{\rho}}}, \tag{37}$$

where $\bar{s}$ denotes a constant value of 0.01, $\hat{h}$ and $\bar{g}$ denote random numbers between 0 and 1, ϑ is determined as:

$$\vartheta = \frac{\Gamma(1+\rho)\cdot \sin \mathrm{e}^{\frac{\pi\rho}{2}}}{\Gamma\frac{1+\rho}{2}\cdot \rho \cdot 2^{\frac{\rho-1}{2}}}, \tag{38}$$

where ρ denotes a constant value of 1.5. In Eq. (36), $\bar{u}$ and $\bar{v}$ present the spiral shape in the search, as determined in Eqs. (39), (40).

$$\bar{u} = \bar{d}\cos(\theta), \tag{39}$$

$$\bar{v} = \bar{d}\sin(\theta), \tag{40}$$

where:

$$\bar{d} = \bar{d}_1 + \bar{Z}\beta_1, \tag{41}$$

$$\theta = -\psi\beta_1 + \theta_1, \tag{42}$$

$$\theta_1 = \frac{3\pi}{2}. \tag{43}$$

$\bar{Z}$ indicates a low value fixed at 0.00565, β_1 denotes a minor integer of 0.005 and ψ refers to an integer number from 1 to the length of the search area (Dim). $\bar{d}$ and $\bar{d}_1$ assume a value between 1 and 20 for fixing the range of the search cycles. However, as per the proposed OLAO method, $\bar{d}$ and $\bar{d}_1$ are randomly generated with $\tau = 2,414$ as:

$$\bar{x}_{\bar{n}+1} = 2\tau\,|\bar{x}_{\bar{n}}|\,(1 - 2\,|\bar{x}_{\bar{n}}|), \quad 0 < \bar{x}_{\bar{n}} < 1. \tag{44}$$

Step 3. Expanded exploitation (D_3). Whenever the prey area is identified and the Aquila is ready to land and attack, the third method is used D_3. This behavior is represented as:

$$D_3(\tilde{l}+1) = \left[D_{\text{best}}(\tilde{l}) - D_{\tilde{M}}(\tilde{l})\right]\cdot\alpha - \text{rand} + [(UB - LB)\cdot\text{rand} + LB]\times\mu. \tag{45}$$

Here $D_3(\tilde{l}+1)$ denotes the solution of the next iteration $\tilde{l}$, $D_{\text{best}}(\tilde{l})$ indicates the prey's approximate location until the $\tilde{\imath}$-th iteration (the greatest solution), and $D_{\tilde{M}}(\tilde{l})$ signifies the mean value of the current solution at the t-th iteration. α and μ are the exploitation modification parameters set to a minimum value of 0.1. *LB* indicates the problem's lower bound, and *UB* signifies the problem's upper bound.

Step 4. Narrowed exploitation D_4. While the Aquila model prey in the 4-th method D_4, it strikes over land depending on their stochastic motions. Such an approach is referred to as "walk and grab prey". AO attacks the prey at the last location. This behavior is described as:

$$D_4(\tilde{l}+1) = QF\cdot D_{\text{best}}(\tilde{l}) - \left[\tilde{G}_1\cdot D(\tilde{l})\cdot\text{rand}\right] - \tilde{G}_2\cdot\text{Levy}(\beta) + \text{rand}\cdot\tilde{G}_1. \tag{46}$$

$D_4(\tilde{l}+1)$ denotes the solution of the fourth search method's iteration $\tilde{l}$, and *QF* represents a quality function from Eq. (47), used to equalize the search techniques. $\tilde{G}_1$ represents different AO movements that are utilized to track the prey during the flight and is derived using Eq. (48). $\tilde{G}_2$ provides decreasing numbers from 2 to 0, reflecting the AO's flight slope used to follow the prey during its elope from the 1-st to the last $(\tilde{l})$ location, as created by Eq. (49). $D(\tilde{l})$ denotes the present iteration's $\tilde{l}$-th solution.

$$QF(\tilde{l}) = \tilde{l}^{\frac{2\cdot\text{rand}-1}{(1-\tilde{L})^2}}, \tag{47}$$

$$\tilde{G}_1 = 2\cdot\text{rand} - 1, \tag{48}$$

$$\tilde{G}_2 = 2\cdot\left(1 - \frac{\tilde{l}}{\tilde{L}}\right). \tag{49}$$

$QF(\tilde{l})$ is the $\tilde{l}$-th iteration's quality function value, $\tilde{l}$ and $\tilde{L}$ show the current iteration as well as the higher count of iterations. Algorithm 1 illustrates the pseudo-code of the proposed OLAO model.

Algorithm 1. OLAO scheme adopted

Initialization phase

Population initialization D in AO

Initialize the AO parameters

As per the proposed OLAO model the OBL concept is deployed

`while` (the end condition is not met) `do`

Compute the fitness function values:

$D_{\text{best}}(\tilde{l})$

`for` $\tilde{\imath} = 1, 2, \ldots, A$ `do`

Mean value update $D_{\tilde{M}}(\tilde{l})$.

Update $\bar{v}$, $\bar{u}$, $\tilde{G}_1$, $\tilde{G}_2$, Levy(β), etc.

`if` $\tilde{l} \leqslant \frac{2}{3}\cdot\tilde{L}$ `then`

`if` rand $\leqslant 0.5$ `then`

Step 1. Expanded exploration (D_1)

Current solution update in Eq. (34)

`if` Fitness$[D_1(\tilde{l}+1)] <$ Fitness$[D(\tilde{l})]$ `then`

$D(\tilde{l}) = D_1(\tilde{l}+1)$

`if` Fitness$[D_1(\tilde{l}+1)] <$ Fitness$[D_{\text{best}}(\tilde{l})]$ `then`

$D_{\text{best}}\tilde{l}) = (D_1(\tilde{l}+1)$

`end if`

`end if`

`else`

Step 2. Narrowed exploration (D_2)

Current solution update in Eq. (36)

`if` Fitness$[D_2(\tilde{l}+1)] <$ Fitness$[D(\tilde{l})]$ `then`

$D(\tilde{l}) = D_2(\tilde{l}+1)$

`if` Fitness$[D_2(\tilde{l}+1)] <$ Fitness$[D_{\text{best}}(\tilde{l})]$ `then`

$D_{\text{best}}(\tilde{l}) = D_2(\tilde{l}+1)$

$\bar{d}$ and $\bar{d}_1$

are randomly generated as in Eq. (44)

`end if`

`end if`

`end if`

`else`

`if` rand $\leqslant 0.5$ `then`

Step 3. Expanded exploitation (D_3)

Current solution update in Eq. (45)

`if` Fitness$[D_3(\tilde{l}+1)] <$ Fitness$[D(\tilde{l})]$ `then`

$D(\tilde{l}) = (D_3\tilde{l}+1)$

```
      if Fitness[D3(l̃ + 1)] < Fitness[Dbest(l̃)] then
       Dbest(l̃) = (D3 l̃ + 1)
      end if
     end if
    else
     Step 4. Narrowed exploitation (D4)
     Current solution update in Eq. (46)
     if Fitness(D4[l̃ + 1)] < Fitness[D(l)] then
      D(l̃) = (D4 l̃ + 1)
      if Fitness[D4(l̃ + 1)] < Fitness[Dbest(l̃)] then
       Dbest(l̃) = (D4 l̃ + 1)
      end if
     end if
    end if
   end if
  end for
 end while
Return (Dbest)
```

7. Results and Discussions

The adopted multimodal sarcasm detection with HC + OLAO scheme was implemented in Python. The outcomes were computed for the extant schemes such as HC + PRO [50], HC+AO [43], HC+SSO [51], HC+CMBO [52], HC+ALO [53], CNN [54], RNN [55], RF [56], NB [57], Bi-GRU [26], and NN [23]. Furthermore, its performance was evaluated by varying the learning percentage metrics, such as precision, sensitivity, accuracy, specificity, FNR, FDR, F-score, MCC, FPR, rand index, and NPV, correspondingly.

Next, the authors extracted a representative sample from the collection of 6,365 annotated videos. The dataset obtained contained 690 movies with an equal amount of sarcastic and non-sarcastic classifications. The sample images are shown in Fig. 4.

7.1. Dataset Description

The dataset is taken from [58]. The MUStARD dataset is a multimodal video corpus used for automated sarcasm discovery studies. The Big Bang Theory, The Golden Girls, Friends, and Sarcasmaholics Anonymous are just a few of the well-known TV programs that are included in the dataset. MUStARD is a collection of sarcastic label-annotated audiovisual utterances accompanied by their context, which offers more details about the situation in which the utterance is made. A novel dataset (MUStARD) comprises short videos that have been carefully annotated for their sarcastic feature, allowing researchers to investigate the topic. They chose to work with a balanced sample of sardonic as well as non-sarcastic clips to enable us to conduct our tests that expressly focus on the multimodal components of sarcasm.

7.2. Performance Analysis

The performance analysis of the presented HC+OLAO model is illustrated in Figs. 5–7. The adopted HC + OLAO scheme attains higher accuracy (0.86) for the learning rate of 60 percent (compared to the learning rate of 80 percent) than other existing schemes, as shown in Fig. 5a. This demonstrates the impact of the improved features on the text and video data, and the contribution of the optimization strategy to tuning the weights for better training results.

The HC + OLAO scheme attains higher sensitivity (0.98) (for a learning rate of 80 percent) than other extant schemes – see Fig. 5b. The proposed HC + OLAO scheme has shown a maximum precision value, ensuring better performance than other conventional models at the learning rate of 80 percent, as shown in Fig. 5c. This proves the impact of HC which gets trained with the suitable features. As the weights of LSTM are tuned optimally, the proposed HC + OLAO technique paves the way for better detection of the presence of sarcasm from multimodal inputs.

The metrics of the developed HC + OLAO scheme that are worse than those of the traditional approaches, including FPR, FNR, and FDR, are represented in Fig. 6. Similarly, the adopted HC + OLAO model attains the minimum FDR value for a learning rate of 80 percent, when compared with the learning rate of 80 percent, as shown in Fig. 6c. The HC + OLAO model proves that the lower FPR value offers better performance for the learning rate of 60 percent than the conventional models, as shown in Fig. 6b. The lower FNR (0.2) value of the proposed model means it is less prone to outcome errors at the learning rate of 70 percent, as depicted in Fig. 6a. The performance analysis has proven that the HC + OLAO scheme has converged with the objective (lower error).

Figure 7 represents other metrics analyzed, such as MCC, NPV, rand index, and F-score. The graph clearly illustrates that MCC of the HC + OLAO model attains a higher value (0.71) for learning the learning rate of 70 percent. However, existing models attain lower values, as shown in Fig. 7c. Similarly, the proposed model achieves the maximum NPV value (0.8) for a learning rate of 60 percent, compared to the learning rate of 80 percent, as shown in Fig. 7a. Likewise, the F-score for the learning rate of 70 percent is superior to other traditional schemes (Fig. 7b). The rand index for the learning rate of 60 percent achieves a higher value (0.99). Consequently, it has been proven that the presented HC + OLAO model surpasses other solutions in terms of performance.

7.3. Overall Performance Analysis

The overall performance analysis of the developed HC + OLAO scheme, comparing it with other models, is summarized in Tables 3–5 for learning rates of 60, 70 and 80 percent, respectively. The learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration. The proposed scheme achieves maximum accuracy values (0.86) to the extant approaches at the learning rate of 60 percent, and superior F-measure outcomes for the learn-

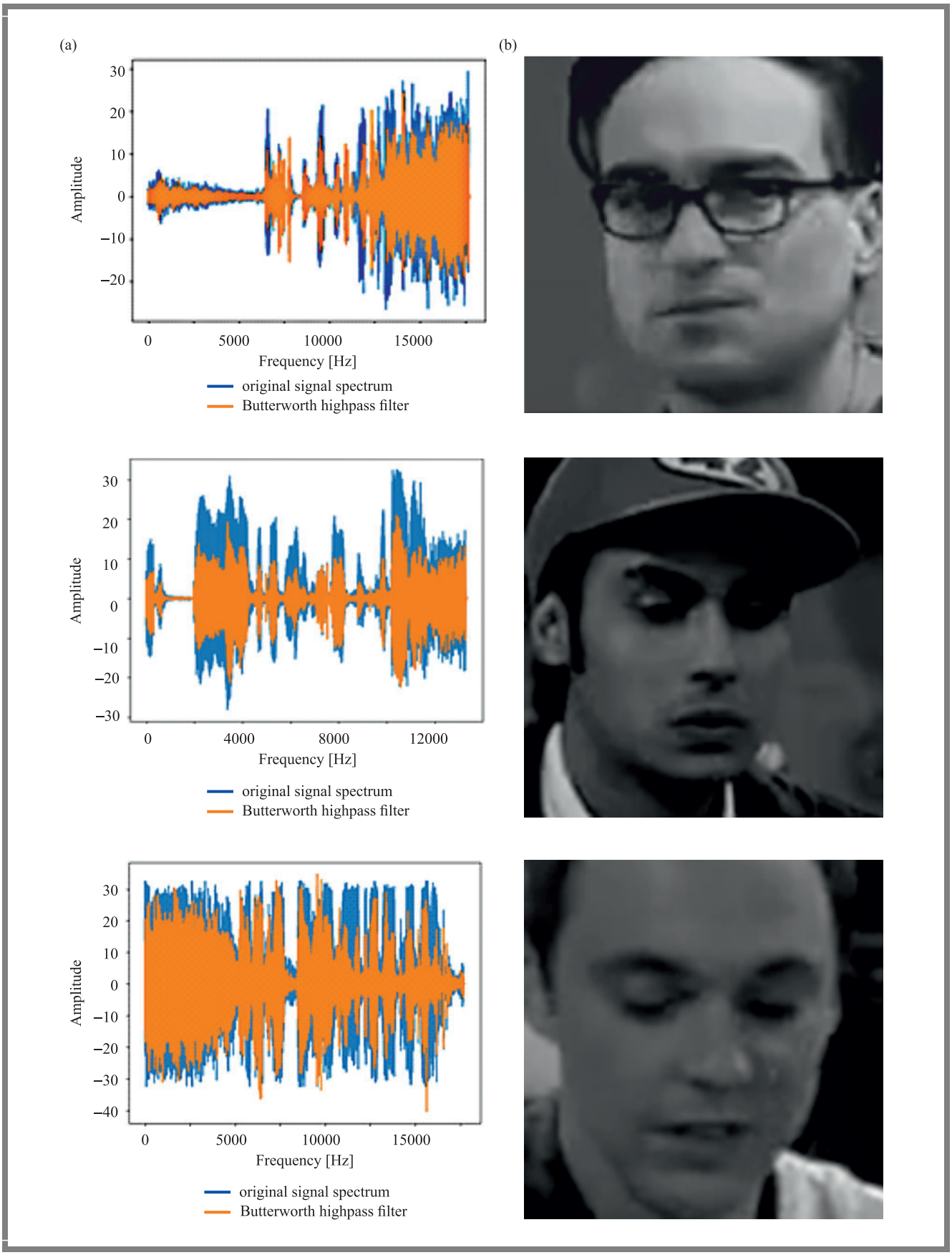

Fig. 4. Representation of: (a) audio preprocessing and (b) image preprocessing.

Tab. 3. Overall performance analysis for the learning rate of 60 percent.

Metric	Method											
	HC + PRO [50]	HC + AO [43]	HC + SSO [51]	HC + CMBO [52]	HC + ALO [53]	CNN [54]	RNN [55]	RF [56]	NB [54]	Bi-GRU [26]	NN [23]	HC + OLAO
FDR	0.40	0.41	0.30	0.45	0.47	0.15	0.21	0.45	0.21	0.33	0.49	0.11
Sensitivity	0.74	0.83	0.75	0.75	0.95	0.56	0.38	0.78	0.76	0.80	0.77	0.79
MCC	0.26	0.29	0.43	0.17	0.23	0.57	0.42	0.19	0.68	0.43	0.09	0.70
Precision	0.62	0.62	0.72	0.57	0.56	0.96	0.92	0.58	0.89	0.70	0.55	0.92
FPR	0.48	0.55	0.32	0.59	0.78	0.13	0.13	0.59	0.27	0.37	0.69	0.09
F-measure	0.68	0.71	0.74	0.65	0.71	0.71	0.54	0.66	0.82	0.75	0.64	0.85
Specificity	0.55	0.48	0.71	0.44	0.25	0.99	0.99	0.44	0.91	0.66	0.34	0.94
FNR	0.29	0.40	0.28	0.28	0.25	0.47	0.65	0.25	0.24	0.23	0.26	0.24
NPV	0.68	0.73	0.75	0.63	0.78	0.70	0.63	0.66	0.80	0.77	0.59	0.82
Accuracy	0.61	0.60	0.73	0.59	0.60	0.77	0.69	0.61	0.84	0.73	0.56	0.86
Rand index	0.82	0.82	0.87	0.78	0.78	0.89	0.84	0.79	0.92	0.87	0.76	0.94

Tab. 4. Overall performance analysis for the learning rate of 70 percent.

Metric	Method											
	HC + PRO [50]	HC + AO [43]	HC + SSO [51]	HC + CMBO [52]	HC + ALO [53]	CNN [54]	RNN [55]	RF [56]	NB [54]	Bi-GRU [26]	NN [23]	HC + OLAO
FDR	0.16	0.19	0.17	0.19	0.13	0.19	0.36	0.22	0.30	0.25	0.54	0.13
Sensitivity	0.96	0.95	0.92	0.27	0.69	0.36	0.72	0.23	0.81	0.86	0.67	0.83
MCC	0.26	0.24	0.23	0.37	0.61	0.49	0.64	0.33	0.70	0.66	0.02	0.72
Precision	0.57	0.56	0.57	0.97	0.90	1.03	0.91	0.96	0.88	0.83	0.49	0.91
FPR	0.77	0.77	0.74	0.15	0.29	0.14	0.28	0.15	0.33	0.21	0.65	0.11
F-measure	0.71	0.71	0.70	0.42	0.78	0.54	0.80	0.37	0.84	0.85	0.57	0.87
Specificity	0.26	0.26	0.29	1.01	0.94	1.03	0.94	1.01	0.89	0.83	0.38	0.92
FNR	0.21	0.25	0.34	0.76	0.34	0.67	0.31	0.80	0.39	0.34	0.36	0.20
NPV	0.81	0.78	0.74	0.59	0.76	0.65	0.78	0.58	0.82	0.86	0.56	0.85
Accuracy	0.61	0.60	0.60	0.64	0.81	0.72	0.83	0.63	0.85	0.84	0.52	0.87
Rand index	0.79	0.79	0.79	0.81	0.92	0.86	0.92	0.80	0.92	0.93	0.73	0.95

Tab. 5. Overall performance analysis for the learning rate of 80 percent.

Metrics	Methods											
	HC + PRO [50]	HC + AO [43]	HC + SSO [51]	HC + CMBO [52]	HC + ALO [53]	CNN [54]	RNN [55]	RF [56]	NB [54]	Bi-GRU [26]	NN [23]	HC + OLAO
FDR	0.42	0.40	0.22	0.38	0.22	0.33	0.20	0.20	0.36	0.38	0.48	0.19
Sensitivity	0.88	0.91	0.17	0.89	0.17	0.16	0.32	0.26	0.70	0.92	0.56	0.72
MCC	0.30	0.36	0.26	0.38	0.26	0.29	0.40	0.35	0.55	0.42	0.47	0.57
Precision	0.61	0.63	0.92	0.65	0.92	1.03	0.95	0.95	0.82	0.65	0.56	0.84
FPR	0.61	0.59	0.17	0.53	0.17	0.25	0.21	0.17	0.62	0.53	0.56	0.16
F-measure	0.72	0.74	0.29	0.75	0.29	0.28	0.48	0.40	0.76	0.76	0.56	0.78
Specificity	0.42	0.44	1.01	0.50	1.01	1.03	1.00	1.01	0.84	0.50	0.47	0.87
FNR	0.35	0.36	0.86	0.32	0.86	0.87	0.71	0.77	0.67	0.33	0.48	0.31
NPV	0.75	0.80	0.56	0.80	0.56	0.52	0.60	0.58	0.74	0.84	0.47	0.76
Accuracy	0.65	0.68	0.59	0.69	0.59	0.56	0.66	0.63	0.77	0.71	0.52	0.80
Rand index	0.82	0.83	0.78	0.85	0.78	0.76	0.83	0.81	0.88	0.85	0.73	0.91

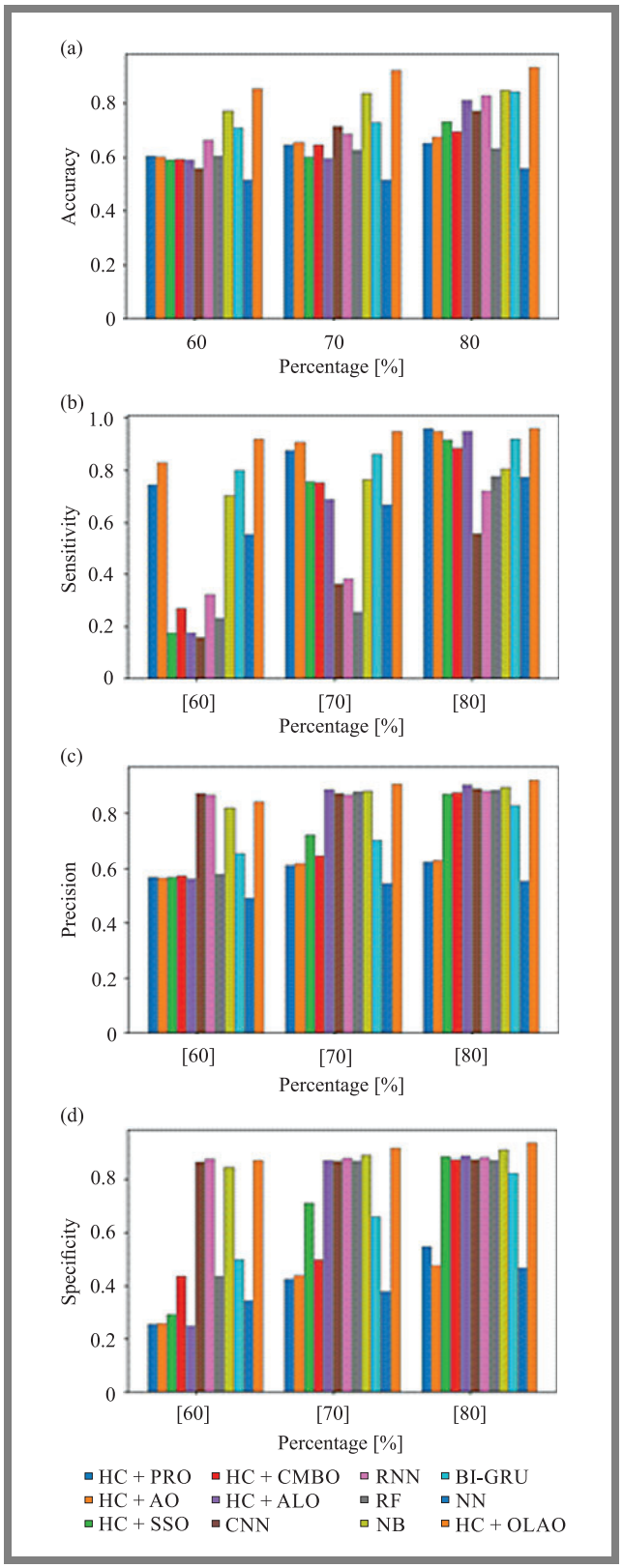

Fig. 5. Performance analysis of the adopted scheme to the extant approaches for: (a) accuracy, (b) sensitivity, (c) precision, and (d) specificity.

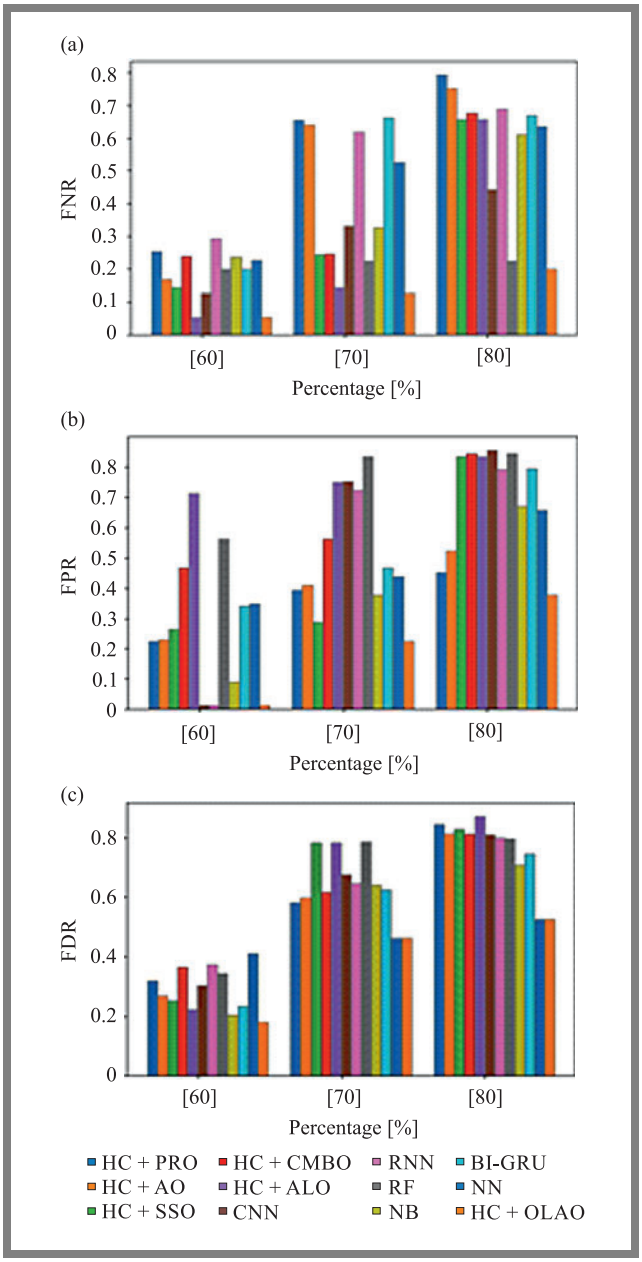

Fig. 6. Performance analysis of the adopted scheme to the traditional approaches for: (a) FNR, (b) FPR, and (c) FDR.

Tab. 6. Statistical analysis with respect to accuracy.

Metric	Std Dev.	Mean	Median	Best	Worst
HC + PRO [50]	0	1.21	1.21	1.21	1.21
HC + AO [43]	0.01	1.17	1.17	1.19	1.16
HC + SSO [51]	0.03	1.20	1.19	1.31	1.19
HC + CMBO [52]	0	1.32	1.32	1.32	1.32
HC + ALO [53]	0.04	1.18	1.16	1.26	1.16
HC + OLAO	0.01	1.16	1.15	1.21	1.15

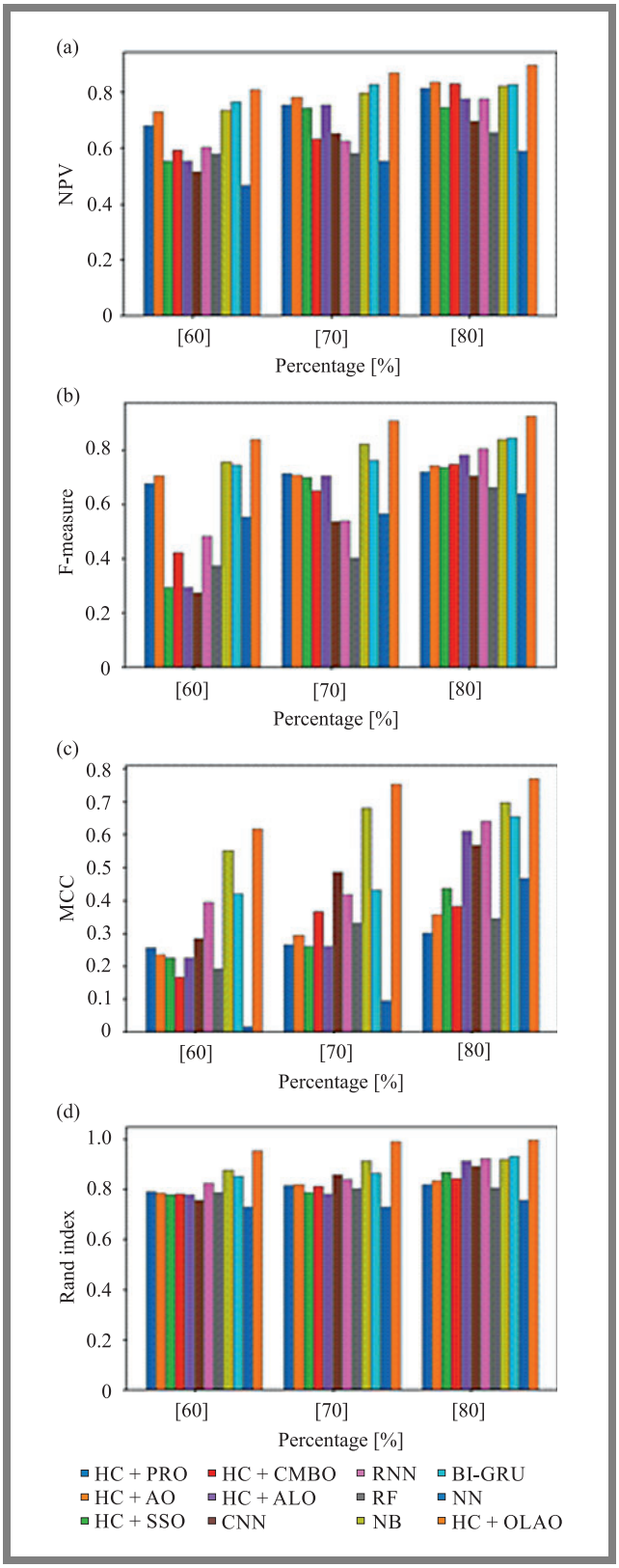

Fig. 7. Performance analysis of the adopted scheme to the traditional approaches for: (a) NPV, (b) F-measure, (c) MCC, and (d) rand index.

ing rate of 70 percent. However, the existing models show the worst performance, as they suffer from lower convergence speed for error minimization purposes.

7.4. Statistical Analysis

The statistical analysis of the proposed approach versus the existing scheme, based on the accuracy metric, is presented in Table 5. The best-case scenario proves an enhancement of the accuracy of results achieved by the proposed HC + OLAO model (1.21), with the said results surpassing the values obtained with the use of other models. Mean performance shows better outcomes for accuracy-related metrics. Therefore, the proposed model has proved to be more effective in multimodal sarcasm detection, almost in all scenarios.

7.5. Features Analysis

The feature-based analysis of the proposed model, with an without relevant comparisons, is illustrated in Table 7.

Also in this case the proposed HC+OLAO model offers better accuracy than the with conventional BoW, without optimization, model with conventional SLBT, and without feature level fusion, respectively. Further, the proposed HC+OLAO model holds lower FNR (0.24) with better performance. This im-

Tab. 7. Analysis based on features type of proposed model.

Metric	Without optimization	With conventional BoW	With conventional SLBT	Without feature level fusion	HC + OLAO
Accuracy	0.62	0.85	0.75	0.70	0.86
Sensitivity	0.80	0.77	0.82	0.97	0.79
Specificity	0.45	0.92	0.68	0.43	0.94
Precision	0.60	0.90	0.72	0.69	0.92
FNR	0.26	0.24	0.24	0.32	0.24
F-measure	0.68	0.83	0.77	0.81	0.85
MCC	0.20	0.69	0.45	0.12	0.70
FPR	0.61	0.09	0.38	0.87	0.09
NPV	0.67	0.80	0.79	0.74	0.82
FDR	0.46	0.10	0.34	0.61	0.11
Rand	0.81	0.92	0.89	0.96	0.94

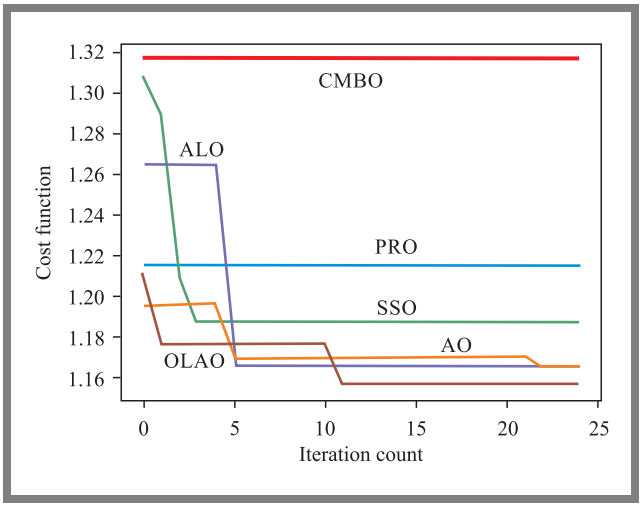

Fig. 8. Convergence analysis of the proposed and other approaches.

plies that the combination proposed in the system is suitable for multimodal sarcasm detection.

7.6. Convergence Analysis

The convergence of the adopted OLAO model is examined and compared with that of the traditional schemes by varying the iteration count between 0, 5, 10, 1, 20, and 25, respectively. Figure 8 illustrates the convergence analysis of the presented method, compared with the traditional schemes. The cost function of the OLAO model is minimized as the count of iterations increases. In addition, the cost function began to decrease from 10–12 iterations. The cost function provides a lower constant value (1.15) for 12–25 iterations than other existing models, such as PRO, AO, SSO, CMBO, and ALO. The proposed OLAO approach achieves the minimum cost function as per the objectives defined in Eq. (27).

Therefore, it is proven that the adopted OLAO approach returns a lower cost function with superior outcomes.

8. Conclusion

This work has identified a new multimodal sarcasm detection method that includes four stages: pre-processing, feature extraction, feature level fusion, and classification. The extracted features were subjected to feature level fusion. In this phase, an improved multilevel CCA fusion technique was applied. The classification was performed using HC solutions, such as LSTM and Bi-GRU. Finally, the outputs of LSTM and Bi-GRU were averaged to obtain an effective output. In order to render the detection method more accurate and precise, the weight of LSTM was tuned using the proposed OLAO model. The final result showed whether any sarcasm was present or not in the analyzed sample. Finally, the results of the adopted technique were compared with the extant methods, with various metrics, including F-score, FDR, specificity, FPR, accuracy, FNR, sensitivity, precision, NPV, rand index, and MCC, taken into consideration. The mean performance of the adopted HC + OLAO approach is better in terms of accuracy-related metrics, when compared with traditional schemes, such as HC + PRO, HC + AO, HC + SSO, HC + CMBO, and HC + ALO.

References

[1] 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*, vol. 12, pp. 6801–6810, 2021 (DOI: 10.1007/s12652-020-02315-1).

[2] Y. Kumar and N. Goel, "AI-Based Learning Techniques for Sarcasm Detection of Social Media Tweets: State-of-the-Art Survey", *SN Comput. Sci.*, vol. 1, no. 6, 2020, (DOI: 10.1007/s42979-020-00336-3).

[3] A. Banerjee, M. Bhattacharjee, K. Ghosh *et al.*, "Synthetic minority oversampling in addressing imbalanced sarcasm detection in social media", *Multimed. Tools Appl.*, vol. 79, pp. 35995–36031, 2020 (DOI: 10.1007/s11042-020-09138-4).

[4] R. Justo, J.M. Alcaide, M.I. Torres *et al.*, "Detection of Sarcasm and Nastiness: New Resources for Spanish Language", *Cogn. Comput.*, vol. 10, pp. 1135–1151, 2018 (DOI: 10.1007/s12559-018-9578-5).

[5] R.A. Potamias, G. Siolas, and A. Stafylopatis "A transformer-based approach to irony and sarcasm detection", *Neural Comput. & Applic.*, vol. 32, pp. 17309–17320, 2020 (DOI: 10.1007/s00521-020-05102-3).

[6] Y. Du, T. Li, M.S. Pathan *et al.*, "An Effective Sarcasm Detection Approach Based on Sentimental Context and Individual Expression Habits", *Cogn. Comput.*, vol. 14, pp. 78–90, 2021 (DOI: 10.1007/s12559-021-09832-x).

[7] L. Ren, B. Xua, H. Lin, X. Liu, and L. Yang, "Sarcasm Detection with Sentiment Semantics Enhanced Multi-level Memory Network", *Neurocomputing*, vol. 401, pp. 320–326, 2020 (DOI: 10.1016/j.neucom.2020.03.081).

[8] M.S. Razali, A.A. Halin, L.S.Y. Doraisamy, and N.M. Norowi, "Sarcasm Detection Using Deep Learning With Contextual Features", *IEEE Access*, vol. 9, pp. 68609–68618, 2021 (DOI: 10.1109/ACCESS.2021.3076789).

[9] S. Rathod, "Hybrid Metaheuristic Algorithm for Cluster Head Selection in WSN", *Journal of Networking and Communication Systems*, vol. 3, no. 4, 2020 (DOI: 10.46253/jnacs.v3i4.a1).

[10] N.S. Lakshmiprabha and S. Majumder, "Face recognition system invariant to plastic surgery", *12th International Conference on Intelligent Systems Design and Applications (ISDA)*, pp. 258–263, 2012 (DOI: 10.1109/ISDA.2012.6416547).

[11] A. Onan and M.A. Toçoğlu, "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).

[12] Meherkandukuri, "Deep Convolutional Neural Network for Emotion Recognition via EEG Signal", *Journal of Computational Mechanics, Power System and Control*, vol. 4, no. 2, 2021 (DOI: 10.46253/jcmps.v4i2.a3).

[13] S. Rajeyyagari, "Automatic speaker diarization using deep LSTM in audio lecturing of e-Khool platform", *Journal of Networking and Communication Systems*, vol. 3, no. 4, 2020 (DOI: 10.46253/jnacs.v3i4.a3).

[14] J. Russel Fernandis, "ALOA: Ant Lion Optimization Algorithm-based Deep Learning for Breast Cancer Classification", *Multimedia Research*, vol. 4, no. 1, (DOI: 10.46253/j.mr.v4i1.a5).

[15] 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).

[16] Y. Diao, *et al.*, "A Multi-Dimension Question Answering Network for Sarcasm Detection", *IEEE Access*, vol. 8, pp. 135152–135161, 2020 (DOI: 10.1109/ACCESS.2020.2967095).

[17] A. Kumar, V.T. Narapareddy, V. Aditya Srikanth, A. Malapati, and L.B.M. Neti, "Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM", *IEEE Access*, vol. 8, pp. 6388–6397, 2020 (DOI: 10.1109/ACCESS.2019.2963630).

[18] Y. Zhang *et al.*, "CFN: A Complex-Valued Fuzzy Network for Sarcasm Detection in Conversations", *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 12, pp. 3696–3710, 2021 (DOI: 10.1109/TFUZZ.2021.3072492).

[19] K. Rothermich, A. Ogunlana, and N. Jaworska, "Change in humor and sarcasm use based on anxiety and depression symptom severity during the COVID-19 pandemic", *Journal of Psychiatric Research*, vol. 140, pp. 95–100, 2021 (DOI: 10.1016/j.jpsychires.2021.05.027).

[20] P. Parameswaran, A. Trotman, and D. Eyers, "Detecting the target of sarcasm is hard: Really?", *Information Processing and Management*, vol. 58, no. 4, 2021 (DOI: 10.1016/j.ipm.2021.102599).

[21] N.Z.Z. Wang, "The paradox of sarcasm: Theory of mind and sarcasm use in adults", *Personality and Individual Differences*, vol. 163, 2020 (DOI: 10.1016/j.paid.2020.110035).

[22] R. Pandey, A. Kumar, J.P. Singh, and S. Tripathi, "Hybrid attention-based Long Short-Term Memory network for sarcasm identification", *Applied Soft Computing*, vol. 106, 2021 (DOI: 10.1016/j.asoc.2021.107348).

[23] N. Basavaraj 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).

[24] D. Jain, A. Kumar, and G. Garg, "Sarcasm detection in mash-

up language using soft-attention based bi-directional LSTM and feature-rich CNN", *Applied Soft Computing*, vol. 91, 2020 (DOI: 10.1016/j.asoc.2020.106198).

[25] Y. Wu *et al.*, "Modeling Incongruity between Modalities for Multimodal Sarcasm Detection", *IEEE MultiMedia*, vol. 28, no. 2, pp. 86–95, 2021, (DOI: 10.1109/MMUL.2021.3069097).

[26] A. Kamal and M. Abulaish "CAT-BiGRU: Convolution and Attention with Bi-Directional Gated Recurrent Unit for Self-Deprecating Sarcasm Detection", *Cogn. Comput.*, vol. 14, pp. 91–109, 2022 (DOI: 10.1007/s12559-021-09821-0).

[27] C.I. Eke, A.A. Norman, S. Liyana, and H.F. Nweke, "Sarcasm identification in textual data: systematic review, research challenges and open directions", *Artif. Intell. Rev.*, vol. 53, pp. 4215–4258, 2020 (DOI: 10.1007/s10462-019-09791-8).

[28] A. Kumar and G. Garg, "Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets", *Journal of Ambient Intelligence and Humanized Computing*, 2019 (DOI: 10.1007/s12652-019-01419-7).

[29] L. Ren, H. Lin, B. Xu, *et al.*, "Learning to capture contrast in sarcasm with contextual dual-view attention network", *Int. J. Mach. Learn. and Cyber.* vol. 12, pp. 2607–2615, 2021 (DOI: 10.1007/s13042-021-01344-2).

[30] Z.L. Chia, M. Ptaszyński, and M. Wroczyński, "Machine Learning and feature engineering-based study into sarcasm and irony classification with application to cyberbullying detection", *Information Processing and Management*, vol. 58, no. 4, 2021, (DOI: 10.1016/j.ipm.2021.102600).

[31] A.F. Hidayatullah and M.R. Ma'arif, "Pre-processing Tasks in Indonesian Twitter Messages", *Journal of Physics: Conference Series*, vol. 801, 2017 (DOI: 10.1088/1742-6596/801/1/012072).

[32] N. Hazim Barnouti, *et al.*, "Face Detection and Recognition Using Viola-Jones with PCA-LDA and Square Euclidean Distance", *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 5, 2016 (DOI: 10.14569/IJACSA.2016.070550).

[33] H. Pandey and R. Tiwari, "An Innovative Design Approach of Butterworth Filter for Noise Reduction in ECG Signal Processing based Applications", *Progress In Science in Engineering Research Journal PISER 12*, vol. 2, pp. 332–337, 2014.

[34] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec", *Information Sciences*, vol. 477, pp. 15–29, 2019 (DOI: 10.1016/j.ins.2018.10.006).

[35] C. Cheng, L. Chunping, H. Yan, and Y. Zhu, "A semi-supervised deep learning image caption model based on Pseudo Label and N-gram", *International Journal of Approximate Reasoning*, vol. 131, pp. 93–107, 2021 (DOI: 10.1016/j.ijar.2020.12.016).

[36] D. Cristinacce and T. Cootes, "Automatic feature localisation with constrained local models", *Pattern Recognition*, vol. 41, no. 10, pp. 3054–3067, 2008 (DOI: 10.1016/j.patcog.2008.01.024).

[37] O.C. Ai, M. Hariharan, S. Yaacob, and L.S. Chee, "Classification of speech dysfluencies with MFCC and LPCC features", *Expert Systems with Applications*, vol. 39, no. 2, pp. 2157–2165, 2012 (DOI: 10.1016/j.eswa.2011.07.065).

[38] T. Kronvall, M. Juhlin, J. Swärd, S.I. Adalbjörnsson, and A. Jakobsson, "Sparse modeling of chroma features", *Signal Processing*, vol. 130, pp. 105–117, 2017 (DOI: 10.1016/j.sigpro.2016.06.020).

[39] M. Kavitha, R. Gayathri, K. Polat, A. Alhudhaif, and F. Alenezi, "Performance evaluation of deep e-CNN with integrated spatial-spectral features in hyperspectral image classification", *Measurement*, vol. 191, 2022 (DOI: 10.1016/j.measurement.2022.110760).

[40] L. An, *et al.*, "Multi-Level Canonical Correlation Analysis for Standard-Dose PET Image Estimation", *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3303–3315, 2016 (DOI: 10.1109/TIP.2016.2567072).

[41] X. Zhou, J. Lin, Z. Zhang, Z. Shao, and H. Liu, "Improved itracker combined with bidirectional long short-term memory for 3D gaze estimation using appearance cues", *Neurocomputing In Press*, vol. 390, pp. 217–25, 2019 (DOI: 10.1016/j.neucom.2019.04.099).

[42] D. Zhao, J. Wang, and Y. Zhang, "Extracting drug–drug interactions with hybrid bidirectional gated recurrent unit and graph convolutional network", *Journal of Biomedical Informatics*, vol. 99, 2019 (DOI: 10.1016/j.jbi.2019.103295).

[43] L. Abualigah, *et al.*, "Aquila Optimizer: A novel meta-heuristic optimization algorithm", *Computers & Industrial Engineering*, vol. 157, 2021 (DOI: 10.1016/j.cie.2021.107250).

[44] B.R. Rajakumar, "Impact of Static and Adaptive Mutation Techniques on Genetic Algorithm", *International Journal of Hybrid Intelligent Systems*, vol. 10, no. 1, pp. 11–22, 2013 (DOI: 10.3233/HIS-120161).

[45] B.R. Rajakumar, "Static and Adaptive Mutation Techniques for Genetic algorithm: A Systematic Comparative Analysis", *International Journal of Computational Science and Engineering*, vol. 8, no. 2, pp. 180–193, 2013 (DOI: 10.1504/IJCSE.2013.053087).

[46] S.M. Swamy, B.R. Rajakumar, and I.R. Valarmathi, "Design of Hybrid Wind and Photovoltaic Power System using Opposition-based Genetic Algorithm with Cauchy Mutation", *IET Chennai Fourth International Conference on Sustainable Energy and Intelligent Systems (SEISCON 2013)*, 2013 (DOI: 10.1049/ic.2013.0361).

[47] A. George and B.R. Rajakumar, "APOGA: An Adaptive Population Pool Size based Genetic Algorithm", *AASRI Procedia*, vol. 4, pp. 288–296, 2013 (DOI: 10.1016/j.aasri.2013.10.043).

[48] B.R. Rajakumar and A. George, "A New Adaptive Mutation Technique for Genetic Algorithm", *In proceedings of IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pp. 1–7, 2012, (DOI: 10.1109/ICCIC.2012.6510293).

[49] F. Chakraborty, P.K. Roy, and D. Nandi, "Oppositional elephant herding optimization with dynamic Cauchy mutation for multilevel image thresholding", *Evol. Intel. 12*, pp. 445–467, 2019 (DOI: 10.1007/s12065-019-00238-1).

[50] S.H.S. Moosavi and V.K. Bardsiri, "Poor and rich optimization algorithm: A new human-based and multi populations algorithm", *Engineering Applications of Artificial Intelligence*, vol. 86, pp. 165–181, 2019 (DOI: 10.1016/j.engappai.2019.08.025).

[51] F. Ahmed, "Social Spider Optimization Algorithm", 2015 (DOI: 10.13140/RG.2.1.4314.5361).

[52] M. Dehghani, Š. Hubálovský, and P, Trojovský, "Cat and Mouse Based Optimizer: A New Nature-Inspired Optimization Algorithm", *Sensors*, vol. 21, no. 15, 2021 (DOI: 10.3390/s21155214).

[53] M.O. Okwu and L.K. Tartibu, "Ant Lion Optimization Algorithm", *Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications. Studies in Computational Intelligence*, vol. 929, 2020 (DOI: 10.1007/978-3-030-61111-8_9).

[54] Y. LeCun, K. Kavukvuoglu, and C. Farabet, "Convolutional networks and applications in vision", *Circuits and Systems, International Symposium on*, pp. 253–256, 2010 (DOI: 10.1109/ISCAS.2010.5537907).

[55] K. Ling-Jing and C.C. Chiu, "Application of integrated recurrent neural network with multivariate adaptive regression splines on SPC-EPC process", *Journal of Manufacturing Systems*, vol. 57, pp. 109–118, 2020 (DOI: 10.1016/j.jmsy.2020.07.020).

[56] Z. Masetic and A. Subasi, "Congestive heart failure detection using random forest classifier", *Computer Methods and Programs in Biomedicine*, vol. 130, pp. 54–64, July 2016 (DOI: 10.1016/j.cmpb.2016.03.020).

[57] P.T. Ilia, "Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size ", *Catena*, vol. 145, pp. 164–179, 2016 (DOI: 10.1016/j.catena.2016.06.004).

[58] –, https://github.com/soujanyaporia/MUStARD.

Dnyaneshwar Madhukar Bavkar is a Research Scholar under the guidance of Dr Ramgopal Kashyap in the Department of Computer Science and Engineering (CSE) at Amity University, Raipur, Chhattisgarh. He received a Bachelor of Engineering (B.E.) from Dr. Babasaheb Ambedkar Marathwada University, Aurangabad and a Master

of Engineering (M.E.) from the University of Mumbai, Maharashtra, India in Computer Science & Engineering (CSE). His research area is Machine learning, Natural Language Processing & Data Analysis.

https://orcid.org/0000-0003-4746-0429

E-mail: dnyaneshwarbavkar@ternaengg.ac.in

Department of Computer Science and Engineering, Amity University, Raipur, Chhattisgarh, India

Ramgopal Kashyap has more than 15 years of teaching experience; his research area is Digital Image Processing, Pattern Recognition, and Machine Learning. He has done B.E. and M.Tech. in Computer Science with Honors. He has filed two patents and authored one international book. He has published more than 40 quality research papers in international journals indexed in Science Citation Index (SCI) and Scopus (Elsevier). He serves as an Associate Editor and Editorial board member for more than 100 Science Citation Index, SCI-E, Scopus indexed Journals. He has also written more than 30 book chapters.

https://orcid.org/0000-0002-5352-1286

E-mail: ram1kashyap@gmail.com

Department of Computer Science and Engineering, Amity University, Raipur, Chhattisgarh, India

Vaishali Khairnar is a Professor and Head of Department of Information Technology at Terna Engineering College, Navi-Mumbai. She has total 21 years of teaching experience. She is Board of Studies Member in Information Technology, University of Mumbai. She has guided many Ph.D. and P.G. studies. Her areas of interest are wireless communication, connected vehicles, VANET, Storage etc. She has published more than 50 plus papers in Scopus journal, springer IEEE etc. She has published 3 patents. She has written and published more than five books under Wiley publication. She has completed one consultancy project and currently working on Research funded project in area of connected vehicles under Department of Science and Technology. She has received Best Research Award in 2021.

https://orcid.org/0000-0002-4867-1263

E-mail: vaishalikhairnar@ternaengg.ac.in

Department of Information Technology, Terna Engineering College, Nerul, Navi Mumbai, India.