I.J. Modern Education and Computer Science, **2025, 6, 111-126**
Published Online on December 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2025.06.08

A Novel Multimodal Sarcasm Detection Methodology with Emotion Recognition Using E-RS-GRU and KLKI-FUZZY Techniques

Ravi Teja Gedela
Department of Computer Science and Engineering, GITAM School of Computer Science and Engineering, GITAM (Deemed to be University), Visakhapatnam, India
Email: rgedela@gitam.edu
ORCID iD: https://orcid.org/0000-0002-4454-0581

J. N. V. R. Swarup Kumar
Department of Computer Science and Engineering, GITAM School of Computer Science and Engineering, GITAM (Deemed to be University), Visakhapatnam, India
Email: sjavvadi2@gitam.edu
ORCID iD: https://orcid.org/0000-0003-2897-2402

Venkateswararao Kuna
Department of Computer Science and Engineering, GITAM School of Computer Science and Engineering, GITAM (Deemed to be University), Visakhapatnam, India
Email: vkuna@gitam.edu
ORCID iD: https://orcid.org/0000-0001-6336-7297

Sasibhushana Rao Pappu*
Department of Computer Science and Engineering, GITAM School of Computer Science and Engineering, GITAM (Deemed to be University), Visakhapatnam, India
Email: spappu@gitam.edu
ORCID iD: https://orcid.org/0000-0003-4397-7607
*Corresponding Author

Received: 11 December 2024; Revised: 25 February, 2025; Accepted: 20 April, 2025; Published: 08 December, 2025

Abstract: Sarcasm, a subtle form of expression, is challenging to detect, especially in modern communication platforms where communication transcends text to encompass videos, images, and audio. Traditional sarcasm detection methods rely solely on textual data and often struggle to capture the nuanced emotional inconsistencies inherent in sarcastic remarks. To overcome these shortcomings, this paper introduces a novel multimodal framework incorporating text, audio, and emoji data for more effective sarcasm detection and emotion classification. A key component of this framework is the Contextualized Semantic Self-Guided BERT (CS-SGBERT) model, which generates efficient word embeddings. Primarily, frequency spectral analysis is performed on the audio data, followed by preprocessing and feature extraction, while text data undergoes preprocessing to extract lexicon and irony features. Meanwhile, emojis are analyzed for polarity scores, which provide a rich set of multimodal features. The fused features are then optimized using the Camberra-based Dingo Optimization Algorithm (C-DOA). The selected features and the embedded words from the preprocessed texts are given to Entropy-based Robust Scaling - Gated Recurrent Units (E-RS-GRU) for detecting sarcasm. Experimental results on the MUStARD dataset show that the proposed E-RS-GRU model achieves an accuracy of 76.65% and F1-score of 76.9%, with a relative improvement of 2.18% over the best-performing baseline and 1.25% over the best-performing state-of-the-art model. Additionally, KLKI-Fuzzy model is proposed for emotion recognition, which dynamically adjusts membership functions through Kullback-Leibler Kriging Interpolation (KLKI), enhancing emotion classification by processing features from all modalities. The KLKI-Fuzzy model exhibits enhanced emotion recognition performance with reduced fuzzification and defuzzification times.

Index Terms: Sarcasm Detection, Emotion Classification, Frequency Spectral Analysis, Feature Fusion, Feature Optimization

1. Introduction

The opinions of others significantly impact our everyday decisions. Such decisions range from buying a laptop, investing money, and choosing educational institutions, influencing many aspects of daily life. Accessing a wide array of global perspectives has become much simpler in the digital realm. Consumers frequent e-commerce platforms, review websites, and social media to gather feedback on the perceived quality of products or services. In addition, people share their experiences and opinions with online communities through blogs and social media platforms [1, 2, 3].

Governments and large businesses use this information to gauge public sentiment on various topics, including branding, products, and political issues. Consequently, interest in sentiment analysis (SA) of online content has surged. SA seeks to determine sentiment polarity by sorting data into positive, neutral, or negative emotions. This analysis helps to understand individual reactions to products, allowing companies to better adjust their strategies to meet customer needs and expectations. Similarly, feedback on government services enables officials to understand the needs of the populace and make crucial decisions [4, 5, 6]. While expressing discontent, constructive feedback is possible and sarcasm serves as a method to express negative feelings using positive language. In rhetorical terms, sarcasm is essentially a form of irony, usually sharp and directed at a particular target [7, 8]. Sarcasm plays a crucial role in our everyday interactions. It is not only used to inflict emotional pain, but often injects humor into the language we use every day. Also, it is inherently implied. For instance, consider the remark, "I love how you are always on time," he said sarcastically as he glanced at his watch after waiting for an hour. Here, the explicit sentiment seems favorable, but the underlying sentiment is clearly negative, marking the statement as sarcastic.

Understanding sarcasm often requires more than just reading the text of a statement, as it typically involves an implicit meaning. Take, for instance, the text-only comment, "I'm always amazed by your quick responses." Although this appears positive, if it is meant sarcastically, detecting sarcasm can be extremely difficult without additional cues. However, the sarcastic intent becomes clear if this comment is delivered in a multimodal context, where one can see the speaker's ironic facial expressions or hear their mocking tone in a video. This multimodal approach significantly aids in the interpretation of sarcastic remarks [9, 10].

Emojis are gaining popularity as they provide a vivid method to express feelings and sentiments. They are particularly useful in clarifying the implicit sentiments and emotions conveyed in communications. Since sarcasm often hinges on the perception of these implicit cues, emojis could be pivotal in determining whether sarcasm is intended in a statement. At times, the underlying sentiment and emotion may not be apparent just from the words spoken. Here, emojis become instrumental in discerning sarcasm [11, 12]. For instance, consider the statement, "I really admire your dedication to being early," delivered with a flat expression and a friendly tone. This could be confusing and the sarcasm is hard to detect, even with visual and auditory cues. However, including an emoji in the statement, "I really admire your dedication to being early 😒," clearly signals the sarcastic intent.

For efficient Sarcasm Detection (SD), neural networks, say Convolutional Neural Networks (CNN) together with Long-Short Term Memory Networks (LSTM), are deployed with the current advancements in Deep Learning (DL) and the availability of text corpora for English. Moreover, in the multitask setting, great success is depicted by those models in which the model is aided by shared knowledge to learn a more general representation of the input text. Nevertheless, the input consecutive word sequences for those deep models are local, yet they might ignore global word occurrence. Many existing systems for SD, whether using text or audio features, have produced suboptimal results due to slow training speeds, neglecting crucial information, and focusing on irrelevant data features.

Hence, to alleviate these drawbacks, this paper proposes an emoji-aware multi-model SD framework with Emotion Recognition (ER) using several specialized modules, such as Adaptive Feature Fusion (AFF), C-DOA, Word Embeddings from CS-SGBERT, E-RS-GRU, and KLKI-Fuzzy. AFF combines various multimodal features, while C-DOA optimizes the feature set by selecting the most relevant features. The word embeddings from the preprocessed text are then fused with the C-DOA features and passed to the E-RS-GRU classifier, which captures sequential dependencies for sarcasm detection. Finally, KLKI-Fuzzy predicts emotions like Surprise, Smile, Sadness, Anger, Fear, and Disgust by adjusting membership functions based on the input data. These modules work synergistically, enhancing predictive accuracy and addressing the complexities of SD and ER. The contributions of the proposed work are as follows.

1. Novel features, such as Lexicon features, Irony features, MFCC, and PLP from both textual and audio data, are introduced to improve the accuracy of the classification process.
2. C-DOA technique is proposed for feature selection, optimizing the feature set by selecting the most relevant features for improved model performance.
3. A novel CS-SGBERT model is proposed to generate efficient word embeddings that address the limitations of traditional word embedding methods.
4. A deep model, E-RS-GRU, is proposed for efficient sarcasm classification, utilizing sequential dependencies and overcoming issues like gradient explosion and overfitting.
5. The KLKI-Fuzzy model is introduced for emotion recognition, dynamically adjusting membership functions using KLKI, thereby enhancing the classification of emotions through multimodal features.

1.1 Outline

Section 2 comprehensively reviews related work on sarcasm detection and emotion recognition, underscoring existing methods and their limitations. Section 3 describes the proposed multimodal framework, describing individual modules such as feature extraction, feature fusion, feature optimization, and classification. Section 4 discusses the dataset used, experimental setup, results, and comparison with baseline models, demonstrating the efficacy of the proposed framework in both sarcasm detection and emotion classification. Finally, Section 5 concludes the paper, summarizing the contributions and suggesting directions for future work.

2. Literature Review

Recent research has increasingly focused on analyzing multimodal data to better understand complex expressions such as sarcasm, sentiment, emotion, and the use of emojis. DL models integrating text, audio, and visual cues have shown promise in improving SD and ER. This section reviews key methodologies in multimodal SD and ER, laying the groundwork for our proposed framework.

2.1 Sarcasm

Detecting sarcasm is a challenging task due to the subtle nature in which sarcastic remarks are often delivered. This complexity makes SD an intriguing area of research, as highlighted by a review of existing studies [13, 14, 15]. Since multimodal data encompasses a richer set of information compared to text alone, it has the potential to offer additional clues to effectively identify sarcasm. Razali et al. [16] performed a comparison of various studies and found a consistent conclusion that multimodal approaches outperform text-only methods in SD. Likewise, Das et al. [17] highlighted that relying solely on word-level information constrained the effectiveness of the system and presented that multimodal approaches are superior to unimodal approaches to detect sarcasm.

In recent times, social media outlets like Instagram, Twitter, etc. have become popular spaces for sarcasm and humor at the expense of others. Schifanella et al. [18] gathered multimodal data from these platforms, combining visual and textual elements as input. Two frameworks have been introduced here: one focuses on leveraging visual semantics, while the second enhances the approach by concatenating both semantics. In a similar way, Cai et al. [19] fused three different modalities: image characteristics, text characteristics, and image attributes, using a hierarchical fusion model to detect sarcasm. Whereas, Wang et al. [20] proposed a 2D Intra-Attention mechanism to explore the relationship between images and words in their image-text model for SD. For textual features, the authors used BERT embeddings, and the ResNet model was used for images.

Castro et al. [21] curated the MUStARD dataset, a novel multimodal dataset for sarcasm that incorporates text, visual and acoustic modalities. They demonstrated that utilizing multimodal data significantly enhances SD compared to using unimodal data. Building on this, Chauhan et al. [9] extended the MUStARD data set by manually annotating it with implicit and explicit sentiments and emotions, acknowledging that sarcasm often involves implicit cues. They proposed two mechanisms based on attention, of which one is “Inter-segment Inter-modal Attention (Ie-Attention)” and another is “Intra-segment Inter-modal Attention (Ia-Attention),” designed to improve the detection of sarcasm, emotion, and sentiment within a multitask framework. In another work, Wu et al. [10] introduced an attention network designed to detect sarcasm by analyzing word-level inconsistencies across different modalities. Ding et al. [22] developed a multilevel learning framework for multimodal SD, which utilizes late fusion and integrates residual connections to enhance detection accuracy.

2.2 Emotion

Emotion detection [23, 24, 25], similar to SA, is a crucial and complex task that can reveal valuable insights for various applications, such as understanding a person's behavior or generating responses in conversational agents. However, relying solely on text may not always suffice for accurate emotion detection. In other words, unimodal approaches can sometimes fall short in capturing the full range of emotions.

A new multimodal dataset called MELD has been introduced to highlight the significance of multimodality and context in detecting emotions within conversations [26]. Likewise, Hazarika et al. [27] introduced an interactive memory network, a model developed to extract multimodal features from videos consisting of conversations. This demonstrates that this approach enhances performance in ER across various classifications. In a survey, Sharma et al. [28] demonstrated that emotions convey more information than words alone in a conversation. The study also illustrated how multimodality helps to accurately identify emotions during conversations.

Emotion analysis is a complex and challenging task, but it also helps to identify other aspects, such as sentiment [29], humor [30], etc. Aziz et al. [31] introduced an architecture known as “Multimodal Transformers Fusion for Desire, Emotion, and Sentiment Analysis (MMTF-DES)” designed for the multimodal human desire comprehension task. This architecture integrates and fine-tunes two cutting-edge pre-trained multimodal transformer models within a single framework. The authors implemented a training approach called multi-sample dropout to improve the generalizability and efficiency of the base model during training. Zhang et al. [32] presented M2Seq2Seq (“an innovative multimodal multitask learning model”) designed for emotion, sentiment and sarcasm recognition, leveraging intra-modal,

inter-modal, and masked outer-modal self-attention mechanisms to integrate contextual and multimodal information effectively. The model features single-level and multi-task multitask learning decoders, which were explored for their potential to enhance performance.

2.3 Gaps in the Literature

Despite considerable progress in multimodal SD and ER, several challenges remain unresolved. First, existing multimodal fusion methods rely on single-stage fusion. Second, while feature selection plays a crucial role in enhancing model performance, there is a lack of advanced optimization techniques for multimodal data, which may lead to suboptimal feature integration. Additionally, while GRU models are generally used for sequential tasks, they often struggle to capture long-range dependencies in complex, multimodal data, such as sarcasm and emotional expressions. Finally, many fuzzy systems used for emotion detection fail to adapt to the dynamic nature of multimodal data, depending on predefined membership functions. Our proposed framework directly addresses these issues by integrating AFF, C-DOA, word embeddings from CS-SGBERT, E-RS-GRU, and KLKI-Fuzzy, making it more efficient in SD and ER.

3. Methodology

This paper proposes an efficient SD and ER using E-RS-GRU and KLKI-Fuzzy techniques. The block diagram of the proposed technique is presented in Figure 1, which comprises several key components: audio, text, emoji processing, feature engineering, and advanced prediction techniques, each explained in this section.

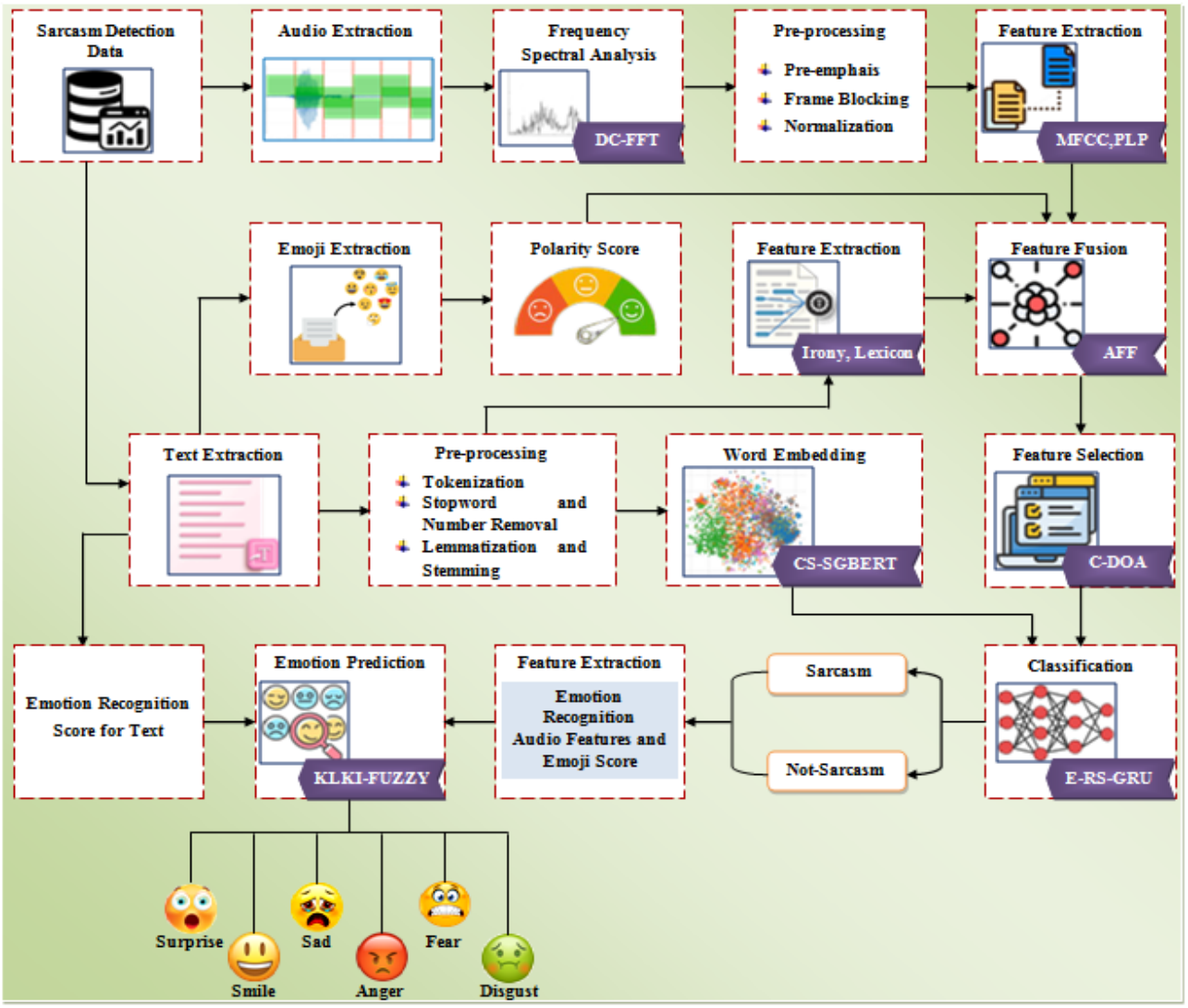

Fig. 1. Proposed Methodology

3.1 Input Data Extraction

The initial step of the proposed architecture begins with extracting audio data (A_d) and text data (T_d) from the SD dataset. Following this, the emojis (E_d) embedded within T_d are identified and extracted. Each extracted emoji is then analyzed to calculate its polarity score, represented as ρ, providing a quantitative measure of its sentiment. Once these elements are extracted, they undergo a preprocessing stage to ensure the data is clean, consistent, and ready for further analysis.

3.2 Multimodal Feature Extraction

This section describes the extraction of appropriate features from both audio and textual modalities, which serve as input to the proposed framework. Each modality undergoes preprocessing followed by efficient feature extraction.

3.2.1 Audio Feature Extraction

Before extracting audio features, the A_d first undergoes frequency spectral analysis and a series of preprocessing steps. The "Discrete Convolution Fast Fourier Transform (DC-FFT)" is used to analyze the amplitude, frequency, and phase of sinusoidal signals in the A_d. Conventional FFT lowers the computational load for waveform analysis, but is limited in handling complex, natural signals. To improve its adaptability, a discrete convolution operation is incorporated in conventional FFT. DC-FFT transforms a signal in the time domain into its frequency domain representation, which is given in Equation 1.

$$X(\varepsilon) = \sum_{n=0}^{N-1} X_n . e^{-2\pi\varepsilon n/N}, \;\; X \in A_d \tag{1}$$

Here, X_n implies the n^{th} audio sample in the time domain, ε denotes the frequency index (wave number), N is the total number of samples, and $X(\varepsilon)$ is the resulting frequency-domain component at ε. A key principle in spectral analysis is that convolution in the time domain corresponds to multiplication in the frequency domain. This property is extremely useful in audio processing because it allows operations like filtering and smoothing to be performed more efficiently in the frequency domain. This is shown in Equation 2.

$$\mathcal{F}\{x[n] * y[n]\} = X(\varepsilon).Y(\varepsilon) \tag{2}$$

Here, $x[n]$ and $y[n]$ are time-domain signals, $*$ denotes convolution, $\mathcal{F}\{.\}$ represents the Fourier transform and $X(\varepsilon)$, $Y(\varepsilon)$ are the corresponding frequency-domain representations. The multiplication on the right-hand side is element-wise. This analysis demonstrates the frequency characteristics that are necessary for identifying sarcasm-related prosodic cues like pitch, tone, and intonation. The resulting signal in the frequency domain is denoted as A_{an}, which is then refined through preprocessing steps such as pre-emphasis, frame blocking, and normalization. From the obtained normalized data, denoted as A_{norm}, two key features are extracted that effectively capture the perceptual characteristics of speech:

1. **Mel-Frequency Cepstral Coefficient (MFCC)**: This is used to represent the short-term power spectrum of speech in a way that aligns with human hearing sensitivity. The human ear is better sensitive to changes in lower frequencies than higher ones, so a nonlinear scale called the Mel scale is used. A_{norm} is first divided into frames and transformed into the frequency domain using the FFT, resulting in a power spectrum with frequency values. Equation 3 describes the conversion from linear frequency f value to Mel scale.

$$Mel(f) = 1125.\ln\left(1 + \frac{f}{700}\right) \tag{3}$$

2. **Perceptual Linear Prediction (PLP)**: It approximates the power spectrum of speech by simulating how humans sense sound, using three essential transformations: intensity-to-loudness mapping (I), critical-band spectral resolution (ζ), and equal-loudness pre-emphasis (ξ). A_{norm} is initially transformed to the frequency domain, and the energy values are grouped into critical bands. In each band k, I_k is weighted using ζ_k and ξ_k. The weighted spectrum is computed using Equation 4.

$$PLP(A_{norm}) = \sum_{k=1}^{K} (\zeta_k . I_k^2 + \xi_k . I_k^2) \tag{4}$$

Both MFCC and PLP are then combined to form the final extracted audio feature set Ω_{aud}, represented in Equation 5.

$$\Omega_{aud} = \{Mel(f), PLP(A_{norm})\} \tag{5}$$

3.2.2 Text Feature Extraction

The text input (T_d) first undergoes a three-step preprocessing pipeline to remove irrelevant or redundant components and normalize the data for feature extraction. These steps include tokenization, stop word and number removal, and lemmatization. After preprocessing, the resulting text, denoted as T_{prep}, is used to extract the lexicon features ($\Im_1$) and irony features ($\Im_2$), which are crucial for SD and ER. $\Im_1$ include word length, word frequency, high-frequency word indicators, orthographic characteristics (such as the use of capitalization and punctuation), phonological traits (such as syllable patterns), and part-of-speech tags. $\Im_2$ include the unigram, bigram, token-pair relationships, polarity windowing, polarity shift detection, and polarity dependency. Together, these features form the final text feature set, represented as $\Omega_{text} = \{\Im_1, \Im_2\}$.

3.3 Feature Fusion

The adaptive feature fusion (AFF) technique fuses the audio features (Ω_{aud}), the text features (Ω_{text}) and the polarity score (ρ). Each input passes via a two-layer convolution within the AFF module, producing three distinctive feature maps, denoted $\delta_1 = conv^2(\Omega_{aud})$, $\delta_2 = conv^2(\Omega_{text})$, and $\delta_3 = conv^2(\rho)$. Here, δ implies the feature map and $conv^2$ signifies the two-layered convolution. The feature maps are then merged through an element-wise summation to form the fused map $\hat{\delta} = \delta_1 + \delta_2 + \delta_3$. Subsequently, global max pooling (G) followed by two fully connected layers ($\wp$) transforms $\hat{\delta}$ into channel-wise tensor $\aleph = \wp^2(G(\hat{\delta}))$. The resulting fused feature tensor $\aleph$ serves as an input for the feature selection phase.

3.4 Feature Selection

The optimized features of $\aleph$ are selected using C-DOA. This phase is mainly used to increase classification accuracy and reduce training time. While the standard DOA provides decent global search abilities, it usually converges slowly and may get stuck in local optima, specifically when handling high-dimensional or sparse multimodal data. Likewise, commonly used approaches like Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) suffer from issues such as early convergence, high computational cost, and slow adaptability when the feature space is noisy or has low variance. To handle these limitations, the proposed approach incorporates Canberra distance into the DOA framework, resulting in C-DOA. Unlike Euclidean distance used in many standard methods, Canberra distance places more emphasis on smaller feature values, making it well-suited for sparse and normalized data, such as features extracted from text, audio, and emojis. This change allows for better discrimination among features and enables a more robust exploration of the search space. Consequently, C-DOA enhances convergence speed, selects better suitable features, and leads to better classifier performance compared to PSO, GA, and standard DOA.

The C-DOA algorithm begins by initializing $\aleph$ as a population of dingoes, each describing a candidate feature subset. The alpha, the most potent member of this population, leads the pack based on a fitness (fit) function determined by the maximum classification accuracy (acc) and is formulated as $fit(\aleph) = max(acc)$. The dingoes search the solution space to locate and converge upon this prey. In feature selection, the prey represents the optimal or near-optimal subset of features that maximizes the classification accuracy. The algorithm iteratively enhances feature subsets through two main phases.

1. **Encircling:** Dingoes estimate the prey's location by calculating the Canberra distance between the prey's position P_{prey} and each dingo's position $P_{\aleph}(i)$, which is equated as:

$$dist = \sum \left| \frac{KP_{prey} - P_{\aleph}(i)}{KP_{prey} + P_{\aleph}(i)} \right| \tag{6}$$

   Here, K is a coefficient vector that controls the search behavior. The dingo's next position is updated according to $P(i+1) = P_{prey}(i) - Q \times dist$. Here, $K = 2u_1$, $Q = 2vu_2 - v$, and u_1, u_2 are random vectors in $[0, 1]$, while v decreases linearly from 3 to 0 over iterations i, which is: $v = 3 - \left[i \times \left(\frac{3}{i_{max}} \right) \right]$.

   This phase effectively narrows the search space by letting dingoes approximate the location of the best feature subset, thus concentrating the search towards optimistic regions with the most appropriate features for SD.
2. **Hunting**: In the hunting phase, the dingo population is categorized into alpha ($\aleph_\alpha$), beta ($\aleph_\beta$), and other members ($\aleph_\gamma$) based on superiority, with the alpha leading the hunt. Each group updates its position relative to the prey by computing the Canberra distance, which is formulated as:

$$dist_j = \sum \left| \frac{K_j \aleph_j - P_{prey}}{K_j \aleph_j + P_{prey}} \right|, \; j \in \{\alpha, \beta, \gamma\} \tag{7}$$

   Here, K_j and $\aleph_j$ represent the coefficient vector and current position of the j^{th} group member, respectively, and P_{prey} is the prey's position. The updated position for each group member is calculated as:

$$P_j^{new} = \left|P_j - Q \times dist_j\right|,\ j \in \{\alpha, \beta, \gamma\} \tag{8}$$

Then, the intensity λ_j of each dingo, indicating its hunting strength and fitness, is computed by:

$$\lambda_j = \log\left(\frac{1}{fit_j - (1Q - 100)} + 1\right),\ j \in \{\alpha, \beta, \gamma\} \tag{9}$$

Here fit_j is the fitness score of the member of group j. These iterative position updates continue until convergence, that is, when dingoes no longer improve their positions, indicating the end of the hunt.

After both phases, the best feature subset ($\aleph_{\text{best}}$) with the best fitness is selected and given to the classifier for final SD.

3.5 Word Embedding

Before classification, the preprocessed texts (T_{prep}) are passed through a Word Embedding (WE) layer to transform them into a numerical form that the classifier can process. This process is done using the CS-SGBERT technique. BERT can be finely tuned with a small portion of labeled data to accomplish specific tasks. However, BERT mainly utilizes its final output layer, while intermediate layers capture essential semantic knowledge. The update gate from the LSTM is incorporated with BERT to capture and pass necessary information across layers to leverage this. Moreover, traditional activation functions may struggle with data with zero activation values. This limitation is overcome using the Softmax GELU (Gaussian Error Linear Unit) activation function.

The CS-SGBERT consists of the following layers: input layer, embedding layer, BERT transformer layer, contextual embedding layer, update gate layer, and output layer. Each layer has a specific role in transforming and refining the input text to derive feature representations that are used for classification.

1. **Input Layer**: T_{prep} is given to the input layer and is represented as a sequence of words $\{w_1, w_2, \dots, w_c\} \in T_{prep}$, where w_c is the final word in the sequence. The words are then tokenized into smaller subword units, describing each token within different classes. These tokens are represented as $tok = [t_1,\ t_2, \dots, t_c]$, where t_c implies the tokenized word at position c. Now, these tokenized words are given to the embedding layer for further processing.
2. **Embedding and BERT transformer Layers**: In the embedding layer, words are described as vectors in a high-dimensional space. This process involves three types of embeddings, namely token embedding, segment embedding, and positional embedding. Token embedding transforms each word/subword into a fixed-dimensional vector, with special tokens $[CLS]$ and $[SEP]$ added at the beginning and end of the sentence, respectively. Segment embedding is used to differentiate between distinct input sequences, such as sentences in a pair. Positional embedding adds information about the position of each token within a sequence, compensating for the transformer's lack of inherent ordering awareness.
   These embeddings are summed together and passed through the BERT transformer encoder. The BERT encoder consists of twelve transformer layers, each with twelve attention heads and millions of parameters. The encoder processes the input to capture contextual information for each token, allowing the model to generate contextual embeddings $\{z_1, z_2, \dots, z_{xx}\}$, where z_i represents the contextual embedding for the i^{th} token in the sequence.
3. **Updated Gate Layer**: The updated gate layer is introduced between neurons to collect semantic knowledge from the intermediate layers of the BERT model. This gate helps to control the information flow into the memory and updates it effectively. The updated gate function λ_{upd} is formulated as:

$$\lambda_{upd} = \tan(S^{each}) * tok \tag{10}$$

   Here, S^{each} implies the semantic information of each layer. This gate also helps distill the information passed between layers, assuring that the model retains essential features for classification.
4. **Output Layer**: The output layer consists of a simple classifier model with a fully connected layer and Softmax GELU activation. The loss ($Loss$) in the classifier output is calculated as $Loss = \frac{1}{2}\ (Tar - \sum S')$. Where, Tar is the target WE score (ground truth) and S' is the embedding of the output word, which is calculated as:

$$S' = \sum\left(1 + \frac{2e^{-\lambda_{hid}}}{\sqrt{\pi}} \int_0^{\frac{v}{\sqrt{2}}} e^{-\lambda_{hid}^2}\ dx\right) \tag{11}$$

   Here, v implies the set of hidden neurons and λ_{hid} is the output of the hidden layer. This formula for S' computes the embedding considering the learned features in the hidden layer and the activation pattern,

guaranteeing that the feature representation is enhanced for classification tasks. This S' is used for classification with the Softmax GELU activation function.

3.6 Sarcasm Detection

The WE features $S'$and the best selected feature subset $\aleph_{\text{best}}$ are given to E-RS-GRU classifier to detect sarcasm. These inputs are fused into a single vector, denoted as φ, allowing the model to learn from both features. GRU's are used because they can handle sequential dependencies in text. However, a familiar problem with it is the explosion of gradients and overfitting. To mitigate these, the Entropy-based Robust Scaling (E-RS) approach is applied to the GRU, assuring that the gradients remain stable during training and the model can generalize nicely. Figure 2 shows the E-RS-GRU architecture.

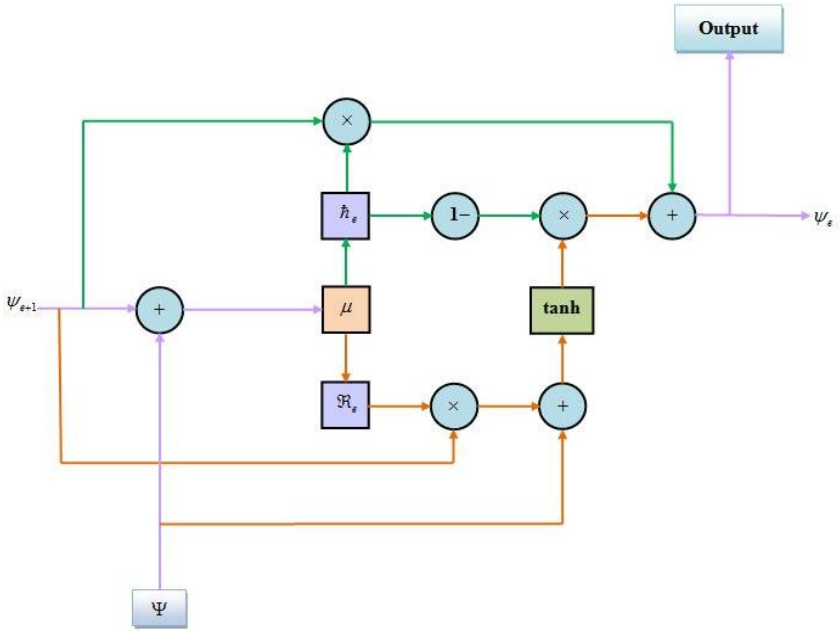

Fig. 2. E-RS-GRU Architecture

The E-RS-GRU model employs two gates, the reset gate and the update gate, to control the information flow via the network. These gates regulate how much previously hidden memory should be retained and how much current input should be incorporated into the memory. By effectively managing this flow, the gates guarantee that the network can retain pertinent information from past inputs while updating its state based on the current context. The operations of each gate is detailed as follows.

1. **Reset Gate**: The reset gate is necessary to control memory retention in the network. It determines how much of the previous hidden memory should be forgotten. In SD, some past information may no longer be appropriate for the current input. The reset gate assures that extrinsic information is discarded, letting the model concentrate on the more pertinent aspects of the current sequence. The reset gate $\Re_e$ is computed as: $\Re_e = \mu(w_{\Re} * \varphi + w_{\Re}\psi_{e-1} + b_{\Re})$. Here, $\Re_e$ is the output of the reset gate at time step e, ψ_{e-1} is the previous hidden memory from time step $e-1$, $w_{\Re}$ is the weight vector associated with the reset gate, $b_{\Re}$ is the bias term, and μ is the E-RS activation function, which is defined in Equation 12:

$$\mu(\Re_e) = \sum \frac{\Re_e - med(\Re_e)}{\Re_e . \log \Re_e} \tag{12}$$

Here, $med(\Re_e)$ signifies the reset gate output median, used to scale the value of the reset gate.

2. **Update Gate**: The update gate controls how much of the previous memory should be memorized for future predictions in the model state. In SD, some past information is crucial to understanding the context, specifically when sarcasm involves referencing earlier parts of the conversation. The update gate ensures that the model preserves useful historical context and effectively incorporates the most relevant information into the current state. The computation of update gate and its activation function is given in Equations 13 and 14, respectively.

$$\hbar = \mu(w_{\hbar} * \varphi + w_{\hbar}\psi_{e-1} + b_{\hbar}) \tag{13}$$

$$\mu(\hbar) = \sum \frac{\hbar_e - med(\hbar_e)}{\hbar_e . \log \hbar_e} \tag{14}$$

Here, $\hbar_e$ is the output of the update gate at time step e, $w_\hbar$ is the weight vector associated with the update gate, ψ_{e-1} is the previous hidden state (memory), and $b_\hbar$ is the bias. Now, the updated memory, $\psi_e = (1 - \hbar_e)\psi_e + \hbar_e \psi_{e-1}$. Where, ψ_e is the updated memory at time step e and $\hbar_e$ is the output from the update gate. Finally, the current memory ψ'_e is computed by passing the input and updated memory through a $tanh$ activation function, which is given in Equation 15.

$$\psi'_e = tanh\left(w_\psi . \varphi + w_{\psi,\psi}(\Re_e * \psi_{e-1}) + b_\psi\right) \tag{15}$$

Here, w_ψ and $w_{\psi,\psi}$ are weight vectors, b_ψ is the bias term, and $\Re_e$ is the output from the reset gate. The update gate thus guarantees that appropriate past information is retained while new information is incorporated to update the memory. This balance between memory retention and adaptation is crucial for SD, where the model must recall and update the context efficiently.

3.7 Emotion Recognition and Prediction

The audio features (Ω_{aud}), the emoji scores (u), and the ER score (τ) of T_d are extracted for emotion recognition. The audio features include the arithmetic mean value, minimum value, maximum value, standard deviation, skewness, variance, kurtosis, zero crossings, median, entropy, moments, mean energy and change in signal values. The extracted feature (σ) set is represented as: $\sigma = \{\Omega_{aud}, u, \tau\}$.

Once σ is acquired, it is given to the KLKI-Fuzzy algorithm for emotion prediction. The algorithm generates rules to forecast emotions such as Surprise, Smile, Sadness, Anger, Fear, and Disgust. Conventional fuzzy systems often use fixed membership functions like triangular, trapezoidal, or Gaussian shapes. While these functions are easy to implement, they are fixed and do not adapt well to the complex and varying nature of real-world multimodal data. This limitation becomes more apparent when multiple modalities (e.g. voice tone and emojis) convey overlapping or ambiguous emotional signals, leading to imprecise rule generation and classification. To address this challenge, the proposed KLKI-Fuzzy model incorporates the KLKI technique into the fuzzy inference system. This permits the model to dynamically adjust the shape and parameters of the membership functions based on the actual input distribution. As a result, the fuzzy system becomes more flexible, adaptive, and capable of capturing subtle variations in emotional expressions. The KLKI-Fuzzy algorithm operates in three stages: fuzzification, rule evaluation, and defuzzification.

1. The **fuzzification** process is the first step in the KLKI-Fuzzy model, where the data is mapped to a set of KLKI membership functions, also referred to as "fuzzy sets." This transformation process is expressed as: $\sigma \xrightarrow{fuzzification} \sigma^g$. Here, σ^g implies the fuzzy output after applying the KLKI function, which is calculated as:

$$\sigma^g = e^{-\frac{1}{2}\sum \sigma \log(\eta) * \frac{\sigma - \tau}{\eta}} \tag{16}$$

Here, τ signifies the center of membership function and η represents the width of the membership function.

2. After fuzzification, the **fuzzy rule evaluation** occurs, where fuzzy sets are assessed based on predefined rules. These rules are formed in the IF-THEN statement form and help to define the relationships between the features and emotions. For this, logical operators such as "and," "or," and "not" are used. The "and" operation is defined as: $\Pi = min(\sigma^g)$. Here, min() indicates the minimum membership value across the fuzzy set. The "or" operation is defined as: $\Theta = max(\sigma^g)$. Here, max() denotes the maximum membership value across the fuzzy set. Following the rule evaluation, the inference engine processes the fuzzy set and maps it to a crisp value, giving an accurate description of the fuzzy set.
3. The next step is the **defuzzification** process, which generates the final output for emotion prediction. The defuzzification process is mathematically expressed as:

$$\chi_{rule} = \frac{\sum_{i=1}^{h} l.\theta}{\sum_{i=1}^{h} \theta} \tag{17}$$

Here, χ_{rule} implies generated rules, h signifies the total number of features, θ delineates the mapped data, and l elucidates membership values for each fuzzy set. In the end, emotion categories such as Surprise, Smile, Sadness, Anger, Fear, and Disgust are predicted by the KLKI-Fuzzy technique based on the generated rules.

4. Results and Discussion

This section presents and analyzes the experimental results, underlining the effectiveness of the proposed approach in comparison to existing methods. All models have been evaluated using the benchmark dataset described in Section

4.1. The details of the experimental setup are provided in Section 4.2, while the performance of the sarcasm detection model is discussed in Section 4.3, and the results for emotion recognition are presented in Section 4.4. The performance of each model is thoroughly discussed and interpreted to provide key insights.

4.1 Dataset Description

The MUStARD dataset [33] a publicly available multimodal resource, is utilized in the proposed work. It comprises 690 videos totaling 9,626 seconds, each annotated with sarcasm labels, including sarcastic and non-sarcastic data points. These videos are sourced from popular television shows such as Friends, The Big Bang Theory, and The Golden Girls, containing audiovisual utterances, facial expressions, speaker information, and sarcasm labels.

The dataset is balanced, equally representing 345 sarcastic and 345 non-sarcastic data points. The distribution of labels across various characters has been carefully designed to avoid speaker bias, including prominent characters like Chandler and Sheldon, and minor characters like Dorothy from The Golden Girls. Each data point is paired with its context, which includes prior dialogue from the same speaker(s), allowing the model to incorporate contextual information for more effective SD. The dataset includes key features such as video, audio, and text transcriptions, all annotated with speaker identifiers. The manual annotation process ensures high inter-annotator agreement, providing a rich source of multimodal data for SD research.

4.2 Experimental Setup

The experiments are conducted using Google Colab, which provides cloud-based GPU resources for efficient deep learning model training. The framework is implemented with Python 3.10, using TensorFlow 2.16 for model development, and Scikit-learn for evaluation metrics. Google Colab's high-performance resources facilitate quicker execution and ease of access, making it an ideal platform for computationally intensive tasks.

Table 1. Hyperparameters

Hyperparameter	**Value**
Optimizer	Adam
Learning Rate	1e-3
Number of Epochs	30
Batch Size	32
Dropout	0.3
ModelCheckpoint	Yes
EarlyStopping	Yes
Patience	5

For the train/test split, the dataset is divided into 80% training and 20% testing, ensuring that the model is trained on sufficient data. The training process includes regularization techniques like dropout to preclude overfitting and enhance model generalization. Hyperparameter tuning is carried out using Hyperopt [34] with cross-validation to determine the optimal values for key hyperparameters. Table 1 details hyperparameter values used in our experiments. For learning rate, values of 0.0001, 0.001, 0.01, and 0.1 were tested, with 0.001 found to be optimal. The values 16, 32, 64, and 128 were tested for batch size, with 32 being the best choice. For the dropout rate, values of 0.2, 0.3, 0.4, and 0.5 were considered, with 0.3 showing optimal. The Adam optimizer with a learning rate of 0.001 was used during training, with 30 epochs. Early stopping was applied with a patience of 5 epochs, which means training was halted if the validation performance did not improve for 5 consecutive epochs.

4.3 Performance Analysis of Sarcasm Detection Model

The performance of the proposed E-RS-GRU model for SD is assessed using several metrics, including accuracy, specificity, sensitivity, False Negative Rate (FNR), False Positive Rate (FPR), precision, recall, and F1-score. These metrics compare the E-RS-GRU model's performance against several baselines, such as CNN, Recurrent Neural Networks (RNN), GRU, and LSTM. The results presented here indicate that incorporating advanced techniques, such as CS-SGBERT for word embeddings and Entropy-based Robust Scaling, significantly enriches SD performance.

As shown in Figure 3, the E-RS-GRU model attains an accuracy of 76.65%, which surpasses the accuracy of 74.47% for GRU, 72.96% for LSTM, 71.84% for RNN, and 70.87% for CNN. The superior performance of E-RS-GRU can be credited to its ability to capture complex patterns in multimodal data effectively. The CS-SGBERT model, integrated into E-RS-GRU, enhances textual feature extraction by generating efficient contextualized word embeddings. These embeddings facilitate the model to capture complex semantic relationships and emotional cues present in the text, which standard static word embedding methods (like fastText, GloVe, or Word2Vec) fail to do. Further, the Entropy-based Robust Scaling addresses challenges common in DL models, such as gradient explosion and overfitting, particularly when dealing with long sequences or multimodal data. By scaling the model's output space and enhancing its stability during training, it is better equipped to manage high-dimensional and complex features.

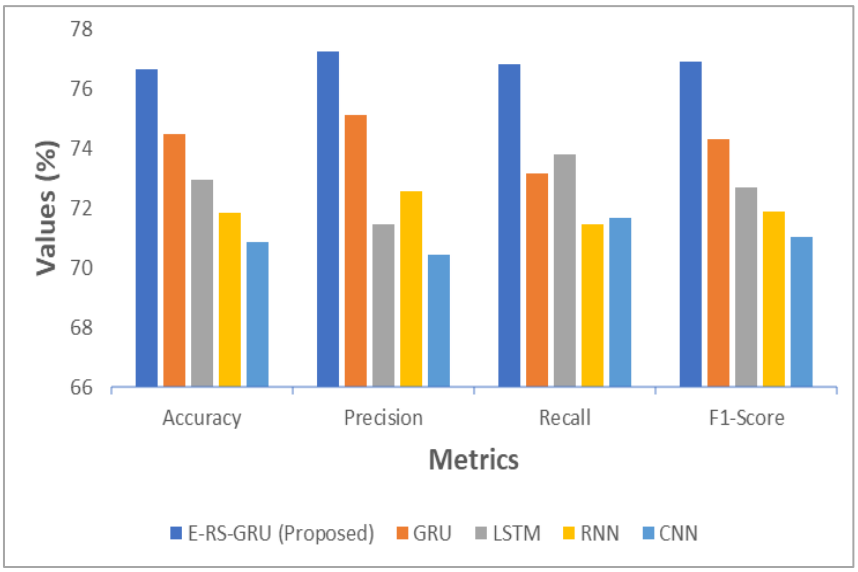

Fig. 3. Performance analysis of the proposed E-RS-GRU

In comparison, DL models such as RNN and CNN encounter limitations in capturing the useful contextual dependencies needed for SD. RNNs tend to struggle with longer sequences, as they constantly fail to understand long-range dependencies efficiently, while CNNs lack the sequential processing knowledge required for understanding temporal context. Although GRU and LSTM can learn sequential dependencies, they still encounter issues such as gradient vanishing and explosion, particularly when handling long text or audio sequences. Moreover, the scarcity of robust feature fusion techniques and dynamic scaling restricts their capacity to model intricate associations in multimodal data. These deficiencies lead to the lower performance of these models when compared to the proposed E-RS-GRU. The E-RS-GRU model also shows an improved precision of 77.25%, recall of 76.56%, and F1-score of 76.9%, demonstrating that it is more proportional in its detection of sarcastic and non-sarcastic data points. These metrics reflect the model's capability to minimize both false negatives and false positives, which is necessary for SD. Improved precision assures that the model does not misclassify too many non-sarcastic instances as sarcastic. At the same time, improved recall means the model can accurately identify most sarcastic instances.

Table 2. Performance analysis (%) of Proposed E-RS-GRU and baseline models

Techniques	Sensitivity	Specificity	FPR	FNR
E-RS-GRU (Proposed)	**76.56**	**77.10**	**22.90**	**23.20**
GRU	73.16	73.47	26.53	26.84
LSTM	73.81	72.93	27.07	26.19
RNN	71.43	71.64	28.36	28.57
CNN	71.67	71.33	28.67	28.33

The performance of the proposed E-RS-GRU method based on sensitivity, specificity, FPR, and FNR is illustrated in Table 2. The sensitivity and specificity values for E-RS-GRU are 76.56% and 77.1%, respectively, indicating significant improvements of 2.75% and 3.63% compared to best-performed baseline models. This suggests that the proposed E-RS-GRU model is more adept at identifying both sarcastic and non-sarcastic data points across a variety of contexts. The gain in sensitivity guarantees that the model is more adequate at identifying sarcastic expressions, while the advancement in specificity confirms that it is better at avoiding false positives in non-sarcastic contexts. Additionally, E-RS-GRU's lower FPR and FNR reveal its ability to avoid false predictions, making it more reliable in SD. This is essential in real-world applications, where false negatives and false positives can greatly influence the accuracy of detection systems. The lower FPR confirms that there are fewer false positives, while the lower FNR means the model is not missing many sarcastic data points.

In summary, E-RS-GRU outperforms baseline models in both accuracy and F1-score, owing to the integration of cutting-edge techniques such as CS-SGBERT for effective contextualized word embeddings and Entropy-based Robust Scaling for model stability. These techniques boost E-RS-GRU to efficiently capture the intricacies of sarcasm in text and audio, providing a robust solution for SD in multimodal data.

4.3.1 Comparison with state-of-the-art (SOTA)

The performance of the proposed E-RS-GRU model is evaluated and compared with several SOTA approaches for SD on the MUStARD dataset, specifically Bhosale et al. [35], Zhang et al. [36], and Chauhan et al. [9]. As shown in

Figure 4, the E-RS-GRU model acquires 76.65% accuracy, 77.25% precision, 76.56% recall, and 76.9% F1-score, which are quite higher than the best-performing SOTA model (Zhang et al. with 75.4% accuracy) by 1.25% in accuracy.

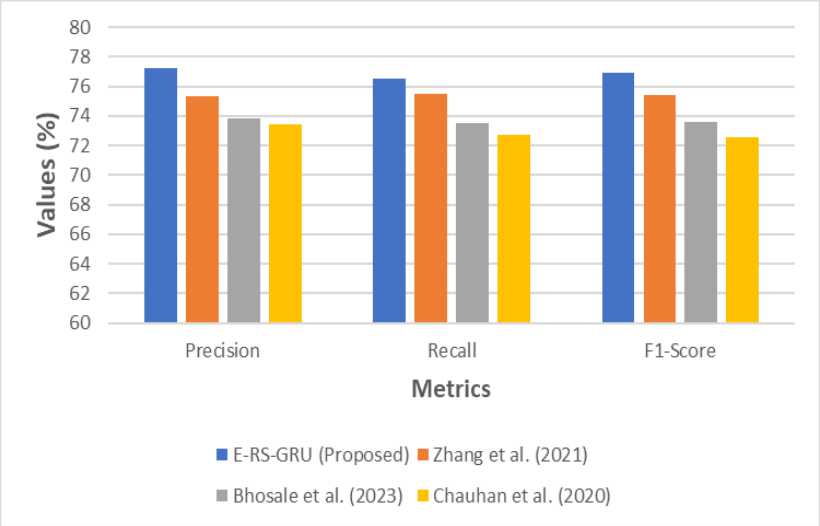

Fig. 4. Comparative Analysis with SOTA Approaches

The enhanced performance of E-RS-GRU can be attributed to the fusion of CS-SGBERT for effective contextualized word embeddings and DC-FFT for more efficient audio feature extraction. These components let the model capture more complex patterns in both textual and audio modalities, improving its ability to detect sarcasm and emotion. The SOTA models performed well but lacked the sophisticated feature extraction and optimization techniques present in the proposed framework, leading to their lower performance.

4.3.2 Ablation Experiments

The ablation experiments were performed to understand how each module contributes to the overall performance of the proposed methodology. We systematically removed key modules, such as contextualized word embeddings from CS-SGBERT, Entropy-based Robust Scaling (E-RS), audio features (MFCC and PLP), textual features (lexicon and irony features), and emoji features, one at a time to monitor how the model's performed varied. When we removed CS-SGBERT from the model, the accuracy was decreased to 73.18%. This suggests that CS-SGBERT, a contextualized WE model, plays an essential role due to its ability to obtain semantic meaning in text data. Similarly, removing E-RS led to a minor yet considerable performance drop, with accuracy decreasing to 74.47%. This E-RS module enables stabilized training by handling gradient explosion and overfitting, which are essential challenges in DL models, mainly when dealing with complex multimodal data. The model is more prone to training instability without this scaling, which can simultaneously degrade performance when learning from text and audio features. This shows the significance of robust scaling techniques, primarily in high data complexity scenarios.

When we excluded the audio features, the accuracy dipped drastically to 70.28%, the most noteworthy decrease observed in the ablation study. This indicates that MFCC and PLP features are required for SD, as they are key prosodic cues that are pretty helpful in contrasting sarcastic speech from non-sarcastic speech. Without them, the model has been learning only from the text and emojis, which are insufficient in order to capture the emotional information in sarcasm. Further, when emoji features are removed, we observed a more moderate performance decline, with an accuracy of 72.46%, demonstrating that emojis are essential in providing emotional context as they convey emotional tone in informal communication, mainly on social media. Their absence leads to losing important emotional cues, reducing the model’s ability to differentiate sarcastic from non-sarcastic content. Lastly, excluding textual features, including lexicon features (e.g., word frequency, word length) and irony features (e.g., token pairs, polarity shifts), led to a performance decrease, bringing accuracy down to 71.73%. These textual features are essential for understanding the linguistic cues and irony that often indicate sarcasm in text. Without these features, the model loses key signals to detect sarcasm accurately.

4.3.3 Statistical Analysis

Paired t-test [37] was conducted for statistical analysis, and the proposed E-RS-GRU model affirms statistically significant improvements over the baseline models, including GRU, LSTM, RNN, and CNN, in terms of accuracy. The t-tests gave a t-value of roughly 6.12 for accuracy and a corresponding p-value of approximately 0.0143. The null hypothesis is rejected with a 0.05 significance level and a p-value below this threshold, demonstrating a considerable performance difference between E-RS-GRU and the other baseline models. These results emphasize the effectiveness of the proposed E-RS-GRU model, showcasing its enhanced ability to detect sarcasm accurately and emphasizing its potential applicability in scenarios where high accuracy is crucial.

4.4 Performance Analysis of the Emotion Recognition Model

Here, the proposed KLKI-fuzzy technique is compared with the existing methods, such as trapezoidal (TZ)-based Fuzzy, triangular (T)-based Fuzzy, Gaussian (G)-based Fuzzy, and fuzzy based on some quality metrics.

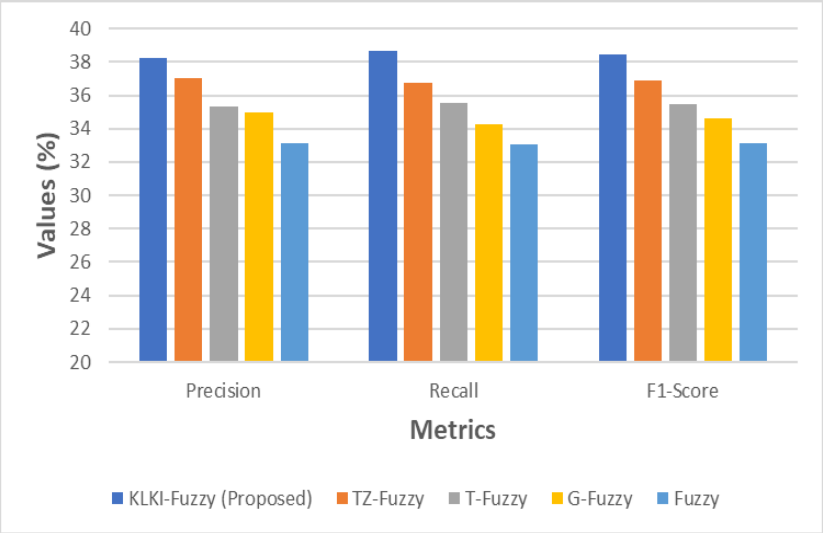

Fig. 5. Performance analysis of Proposed KLKI-fuzzy

In Figure 5, the precision, recall and F1-score of the existing and proposed KLKI-fuzzy models are illustrated. The KLKI-fuzzy obtained a precision of 38.26%, a recall of 38.67%, and an F1-score of 38.46%, whereas the existing models have obtained lower values for both metrics. Due to the modification of the KLKI function, the proposed rule generation model for ER was superior to other techniques.

The time taken for fuzzification, defuzzification and rule generation by the proposed and existing methods is illustrated in Table 3. The proposed has exhibited a decrease of 71, 91, and 1960 ms for fuzzification, defuzzification, and rule generation compared to the existing TZ- fuzzy method. Similarly, the proposed model has obtained a shorter time than all the existing methods. The addition of KLKI in fuzzy has outperformed all other existing activation functions in fuzzy.

Table 3. Performance analysis of Proposed KLKI-fuzzy

Techniques	Fuzzification Time (ms)	Defuzzification Time (ms)	Rule Generation Time (ms)
KLKI-fuzzy (Proposed)	**586**	**576**	**4587**
TZ-fuzzy	657	667	6547
T-fuzzy	712	723	7345
G-fuzzy	832	821	8425
Fuzzy	945	934	9475

5. Conclusion

This work proposes an efficient multimodal SD technique using E-RS-GRU and KLKI-FUZZY. Various steps like frequency spectral analysis, preprocessing, feature extraction, feature fusion, feature selection, sarcasm detection, and emoji prediction are undergone by the proposed work. Next, by employing MUStARD dataset, the performance analysis and the comparative analysis are done to appraise the proposed system's performance. According to the experimental evaluation, the proposed framework achieved accuracy, precision, recall, and F1-score of 76.65%, 77.25%, 76.56%, and 76.9%. With respect to every metric, the proposed model achieved superior performance in comparison to the existing system. Hence, the proposed SD framework is superior to the existing techniques. Nevertheless, this work concentrated only on the multi-model framework, including text, audio, and emoji. Thus, the work will be extended in the future by considering the facial expression of the image along with a novel deep learning technique.

References

[1] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

[2] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal, 5(4):1093–1113, 2014.

[3] Ravi Teja Gedela, Pavani Meesala, Ujwala Baruah, and Badal Soni. Identifying sarcasm using heterogeneous word embeddings: a hybrid and ensemble perspective. Soft Computing, pages 1–14, 2023.

[4] Ronen Feldman. Techniques and applications for sentiment analysis. Communications of the ACM, 56(4):82–89, 2013.

[5] Yanying Mao, Qun Liu, and Yu Zhang. Sentiment analysis methods, applications, and challenges: A systematic literature review. Journal of King Saud University-Computer and Information Sciences, 36(4):102048, 2024.

[6] Karnati Vivek, Sai Mani Garrepalli, PVS Ajay Krishna, Inti Charan Kumar, Ravi Teja Gedela, and Sasibhushara Rao Pappu. Telmemood: A multimodal dataset for sentiment analysis of telugu memes. In 2025 3rd International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC), pages 893–899. IEEE, 2025.

[7] Salvatore Attardo. Irony as relevant inappropriateness. Journal of pragmatics, 32(6):793–826, 2000.

[8] Ravi Teja Gedela, Ujwala Baruah, and Badal Soni. Deep contextualised text representation and learning for sarcasm detection. Arabian Journal for Science and Engineering, 49(3):3719–3734, 2024.

[9] Dushyant Singh Chauhan, SR Dhanush, Asif Ekbal, and Pushpak Bhattacharyya. Sentiment and emotion help sarcasm? a multi-task learning framework for multi-modal sarcasm, sentiment and emotion analysis. In Proceedings of the 58th annual meeting of the association for computational linguistics, pages 4351–4360, 2020.

[10] Yang Wu, Yanyan Zhao, Xin Lu, Bing Qin, Yin Wu, Jian Sheng, and Jinlong Li. Modeling incongruity between modalities for multimodal sarcasm detection. IEEE MultiMedia, 28(2):86–95, 2021.

[11] Dushyant Singh Chauhan, Gopendra Vikram Singh, Aseem Arora, Asif Ekbal, and Pushpak Bhattacharyya. An emoji-aware multitask framework for multimodal sarcasm detection. Knowledge-Based Systems, 257:109924, 2022.

[12] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. arXiv preprint arXiv:1708.00524, 2017.

[13] Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. Automatic sarcasm detection: A survey. ACM Computing Surveys (CSUR), 50(5):1–22, 2017.

[14] GRS Murthy, Ravi Teja Gedela, and Sasibhushana Rao Pappu. Stacking ensemble-based approach for sarcasm identification with multiple contextual word embeddings. In International Conference on Data Management, Analytics & Innovation, pages 71–81. Springer, 2024.

[15] Wangqun Chen, Fuqiang Lin, Guowei Li, and Bo Liu. A survey of automatic sarcasm detection: Fundamental theories, formulation, datasets, detection methods, and opportunities. Neurocomputing, 578:127428, 2024.

[16] Md Saifullah Razali, Alfian Abdul Halin, Noris Mohd Norowi, and Shyamala C. Doraisamy. The importance of multimodality in sarcasm detection for sentiment analysis. In 2017 IEEE 15th Student Conference on Research and Development (SCOReD), pages 56–60, 2017.

[17] Dipto Das. A multimodal approach to sarcasm detection on social media. 2019.

[18] Rossano Schifanella, Paloma De Juan, Joel Tetreault, and Liangliang Cao. Detecting sarcasm in multimodal social platforms. In Proceedings of the 24th ACM international conference on Multimedia, pages 1136–1145, 2016.

[19] Yitao Cai, Huiyu Cai, and Xiaojun Wan. Multi-modal sarcasm detection in twitter with hierarchical fusion model. In Proceedings of the 57th annual meeting of the association for computational linguistics, pages 2506–2515, 2019.

[20] Xinyu Wang, Xiaowen Sun, Tan Yang, and Hongbo Wang. Building a bridge: a method for image-text sarcasm detection without pretraining on image-text data. In Proceedings of the first international workshop on natural language processing beyond text, pages 19–29, 2020.

[21] Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. Towards multimodal sarcasm detection (an _obviously_ perfect paper). arXiv preprint arXiv:1906.01815, 2019.

[22] Ning Ding, Sheng-wei Tian, and Long Yu. A multimodal fusion method for sarcasm detection based on late fusion. Multimedia Tools and Applications, 81(6):8597–8616, 2022.

[23] Pragya Singh Tomar, Kirti Mathur, and Ugrasen Suman. Fusing facial and speech cues for enhanced multimodal emotion recognition. International Journal of Information Technology, 16(3):1397–1405, 2024.

[24] Shiqing Zhang, Yijiao Yang, Chen Chen, Xingnan Zhang, Qingming Leng, and Xiaoming Zhao. Deep learning-based multimodal emotion recognition from audio, visual, and text modalities: A systematic review of recent advancements and future prospects. Expert Systems with Applications, page 121692, 2023.

[25] AV Geetha, T Mala, D Priyanka, and E Uma. Multimodal emotion recognition with deep learning: advancements, challenges, and future directions. Information Fusion, 105:102218, 2024.

[26] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mi- halcea. Meld: A multimodal multi-party dataset for emotion recognition in conversations. arXiv preprint arXiv:1810.02508, 2018.

[27] Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. Icon: Interactive conversational memory network for multimodal emotion detection. In Proceedings of the 2018 conference on empirical methods in natural language processing, pages 2594–2604, 2018.

[28] Garima Sharma and Abhinav Dhall. A survey on automatic multimodal emotion recognition in the wild. Advances in data science: Methodologies and applications, pages 35–64, 2021.

[29] Md Shad Akhtar, Dushyant Chauhan, Deepanway Ghosal, Soujanya Poria, Asif Ekbal, and Pushpak Bhattacharyya. Multi-task learning for multi-modal emotion recognition and sentiment analysis. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 370–379, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[30] Dushyant Singh Chauhan, Dhanush S R, Asif Ekbal, and Pushpak Bhattacharyya. All-in-one: A deep attentive multi-task learning framework for humour, sarcasm, offensive, motivation, and sentiment on memes. In Kam-Fai Wong, Kevin Knight, and Hua Wu, editors, Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 281–290, Suzhou, China, December 2020. Association for Computational Linguistics.

[31] Abdul Aziz, Nihad Karim Chowdhury, Muhammad Ashad Kabir, Abu Nowshed Chy, and Md Jawad Siddique. Mmtf-des: A fusion of multimodal transformer models for desire, emotion, and sentiment analysis of social media data. arXiv preprint arXiv:2310.14143, 2023.
[32] Yazhou Zhang, Jinglin Wang, Yaochen Liu, Lu Rong, Qian Zheng, Dawei Song, Prayag Tiwari, and Jing Qin. A multitask learning model for multimodal sarcasm, sentiment and emotion recognition in conversations. Information Fusion, 93:282–301, 2023.
[33] Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. Towards multimodal sarcasm detection (an _Obviously_ perfect paper). In Anna Korhonen, David Traum, and Lluís Màrquez, editors, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4619–4629, Florence, Italy, July 2019. Association for Computational Linguistics.
[34] James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D Cox. Hyperopt: a python library for model selection and hyperparameter optimization. Computational Science & Discovery, 8(1):014008, 2015.
[35] Swapnil Bhosale, Abhra Chaudhuri, Alex Lee Robert Williams, Divyank Tiwari, Anjan Dutta, Xiatian Zhu, Pushpak Bhattacharyya, and Diptesh Kanojia. Sarcasm in sight and sound: Benchmarking and expansion to improve multimodal sarcasm detection. arXiv preprint arXiv:2310.01430, 2023.
[36] Yazhou Zhang, Yaochen Liu, Qiuchi Li, Prayag Tiwari, Benyou Wang, Yuhua Li, Hari Mohan Pandey, Peng Zhang, and Dawei Song. Cfn: a complex-valued fuzzy network for sarcasm detection in conversations. IEEE Transactions on Fuzzy Systems, 29(12):3696–3710, 2021.
[37] E.C. Hedberg and Stephanie Ayers. The power of a paired t-test with a covariate. Social Science Research, 50:277–291, 2015.

Authors' Profiles

Ravi Teja Gedela obtained his Ph.D. in Computer Science and Engineering from the National Institute of Technology Silchar, Assam, in 2024. He completed his M.Tech and B.Tech in Computer Science and Engineering from JNTU Kakinada in 2012 and 2010, respectively. He works as an Assistant Professor in the Department of Computer Science and Engineering at GITAM (Deemed to be University), Andhra Pradesh, India. Prior to this, he served in various academic roles at reputed engineering institutions. His research focuses on Natural Language Processing, with a special focus on sarcasm detection in multilingual contexts, along with interests in Machine Learning and Deep Learning applications. He has published several SCIE-indexed journal articles and conference papers with Springer and IEEE. Dr. Gedela has also qualified for UGC-NET multiple times and is actively involved in mentoring and collaborative research.

J.N.V.R. Swarup Kumar obtained his Ph.D. in Information Technology from Annamalai University, Tamilnadu, India, in 2022. He received his M.Tech in Information Technology from GITAM University in 2010. He has 15 years of teaching experience at various engineering colleges. Presently, he works as an Assistant Professor in the Department of CSE at GITAM (Deemed to be university). He has over 50 publications in various reputed international journals/conferences, viz., IEEE, Elsevier, and Springer. He also published 7 Patents by IP India Services, Government of India. He is a member of IEEE. His research interests include Data Science, Wireless Sensor Networks, IoT, and Smart Vehicular Networks.

Kuna Venkateswararao is an Assistant Professor in the Department of Computer Science and Engineering at GITAM (Deemed to be) University, Visakhapatnam, India. He received his BE degree and MTech degree in Computer Science and Engineering from Jawaharlal Nehru Technological University, Kakinada, India, in 2014 and 2017, respectively. He completed PhD in the Department of Computer Science and Engineering at the National Institute of Technology Goa. His research interests include Data Science, Wireless Software Defined Networks, 5G Mobile Networks, blockchain and UAV communications.

Sasibhushana Rao Pappu pursuing Ph.D in Computer Science and Engineering (CSE) from JN- TUK, Kakinada, Andhra Pradesh. Master of Technology in CSE in 2012 with first division from JNTU, Vizianagaram and Bachelor of Technology in CSE in 2009 with first division from Raghu En- gineering College (Autonomous). He has about 12 years of work experience in Teaching. Currently he is working as Assistant Professor in CSE department at GITAM (Deemed to be University). He has published papers in various conferences, national & international journals and awarded JRF(UGC- NET).

How to cite this paper: Ravi Teja Gedela, J. N. V. R. Swarup Kumar, Venkateswararao Kuna, Sasibhushana Rao Pappu, "A Novel Multimodal Sarcasm Detection Methodology with Emotion Recognition Using E-RS-GRU and KLKI-FUZZY Techniques", International Journal of Modern Education and Computer Science(IJMECS), Vol.17, No.6, pp. 111-126, 2025. DOI:10.5815/ijmecs.2025.06.08