Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1653696, 10 pages
https://doi.org/10.1155/2022/1653696

Research Article

Multimodal Sarcasm Detection: A Deep Learning Approach

Santosh Kumar Bharti,[1] Rajeev Kumar Gupta,[1] Prashant Kumar Shukla,[2] Wesam Atef Hatamleh,[3] Hussam Tarazi,[4] and Stephen Jeswinde Nuagah[5]

[1]*Pandit Deendayal Energy University, Gandhinagar, Gujarat, India*
[2]*Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, 522502 Andhra Pradesh, India*
[3]*Department of Computer Science, College of Computer and Information Sciences, King Saud University, P.O. Box 51178, Riyadh 11543, Saudi Arabia*
[4]*Department of Computer Science and Informatics, School of Engineering and Computer Science, Oakland University, Rochester Hills, MI, USA*
[5]*Department of Electrical Engineering, Tamale Technical University, Ghana*

Correspondence should be addressed to Stephen Jeswinde Nuagah; jeswinde@tatu.edu.gh

Received 26 February 2022; Revised 23 March 2022; Accepted 28 March 2022; Published 23 May 2022

Academic Editor: Mohammad Farukh Hashmi

In the modern era, posting sarcastic comments on social media became the common trend. Sarcasm is often used by people to taunt or pester others. It is frequently expressed through inflexion, tonal stress in speech or in the form of lexical, pragmatic, and hyperbolic features present in the text. Most of the existing work has been focused on either detecting sarcasm in textual data using text features or audio data using audio features. This article proposed a novel approach by combining textual and audio features together to detecting sarcasm in conversational data. This hybrid method takes a combined vector of extracted audio and text features from their respective models as the input. This combined features will compensated the shortcomings of only text features and vice-versa. The obtained result of hybrid model outperforms both the individual model significantly.

1. Introduction

With the growing number of virtual assistants with voice-to-voice interaction, detecting sarcasm has become crucial task. Artificial intelligence (AI) assistants like Siri, Alexa, and Cortana should be able to identify sarcasm in the user's speech. In 2018, Google used an AI assistant to book an appointment for a haircut by a voice call [1]. So, in such scenarios, it becomes very important that it is able to understand that humans are using sarcastic speech or not. Companies like Amazon and Flipkart rely heavily on customer feedback and reviews in order to improve their products and services. Based on these reviews, they take data-driven decisions, so while analyzing them, it becomes very important that they are able to identify the exact intent behind it. In such scenarios, they should be able to detect whether the given text review is sarcastic or not.

Macmillan English dictionary defines sarcasm as "the activity of saying or writing the opposite of what you mean, or of speaking in a way intended to make someone else feel stupid or show them that you are angry." For example, in this statement "I love the pain present in the breakups," a shift in sentiment can be observed. Even though the sentence tries to convey that one loves the pain present in breakups, but the actual meaning that it tries to convey is the exact opposite. While such patterns are an indication of the presence of sarcasm in a statement, other lexical and pragmatic features as shown by [2, 3] also play an important role in the detection of sarcasm.

In some cases, only text will not suffice to detect sarcasm, and additional cues would be required to detect it accurately. For example the phrase "yeah right" would have different meanings which depends on how the person says it and what is the context behind it [4]. In such cases, factors like changes in tone, overemphasis of words, and drawn-out syllables would play an important role in deciding whether it is sarcastic or not.

The aim of this article is to identify sarcasm in conversational data using a multimodal approach, where textual and audio feature will combined to identify sarcasm. Majority of the work done until date has focused on textual cues for detecting sarcasm, so in this approach, the effect of audio cues while detecting sarcasm is observed. First, a model will be built that works only with features extracted from audio, and then, at a later stage, features from text will also be added. These features are extracted using deep learning techniques rather than performing the task of feature extraction manually. Through this approach, using audio features along with text features gives significantly better results when compared to an approach which only uses textual features.

The rest of the article is organized as follows: Section 2 presents the literature survey on sarcasm detection for textual as well as audio data. The methodologies are given in Section 3. The proposed work is explained in Section 4. Section 5 draws results and analysis of the proposed work. Finally, Section 6 concludes the article, and the abbreviations section depicts all the abbreviations used in the article.

2. Related Work

Most of the work done until the date has been focused on textual cues in order to detect sarcasm as detecting sarcasm from text is very difficult [5–9]. While working with text, researchers used to carry out feature engineering on the dataset in order to detect sarcasm. In early days, sarcasm was detected by observing some particular patterns like positive comment in negative situations [9]. Researchers have also used lexical features like unigram, bigram, and trigram in order to detect sarcasm. The importance of lexical features in detecting sarcasm and irony was first observed by Kreuz et al. [10]. Features like punctuation symbols and interjection also play an important role in recognizing sarcasm [3]. Along with these, features like quotes and intensifiers can be broadly classified into hyperbolic features. Utsumi [11] showed that by using such features, stress can be given on a particular word present in the text, which will act as a cue for the presence of sarcasm. With the help of pragmatic features like emoticons, smilies, and special punctuations, Carvalho et al. [12] detected irony from newspaper text data. The usage of such features would be highly effective while detecting sarcasm from tweets, as they usually contain such features.

However, recently, researchers have left the task of feature engineering on the deep learning models itself rather than performing it manually [13]. While working with these deep learning models, preprocessing is required for the input text. For each word present in the sentence, it is converted into its embedding vector. Such embedding vectors can be generated with the help of pretrained models like GloVe [14]. Mandal et al. [15] achieved an accuracy of 86.6% while detecting sarcasm from news headlines. Their model consisted of a CNN-LSTM neural network with inputs as word embedding vectors of news headlines. Zhang et al. [13] used bi-directional gated recurrent neural networks and discrete models to detect sarcasm. Their results show that the neural models outperform their discrete models while detecting sarcasm from tweets. Researchers have seen an improvement in the accuracy of natural language processing (NLP) tasks while using pretrained word embeddings [16, 17].

Ibrahim et al. [18] introduce transformer-based pretrained for NLP application like sentiment analysis and sarcasm detection in other than English language. Zhang et al. [19] proposed a complex-valued fuzzy network by leveraging the mathematical formalisms of quantum theory and fuzzy logic to resolve the intrinsic vagueness and uncertainty of human language in emotional expression and understanding for sarcasm detection. Sarcasmdet has been introduced for sarcasm detection in Arabic language text using arabert pretrained model [20, 21]. Bedi et al. [22] have developed a benchmark dataset for sarcasm detection in non-English language. Further, they devised MSH-COMICS, a novel attention-rich neural architecture for the utterance classification.

While detecting sarcasm, humans try to look for cues like changes in tone or frequency of the sound (or audio). Researchers have exploited these methods to get the models detect sarcasm from audio. Acoustic features like mean f0, standard deviation of f0, f0 range, mean amplitude, amplitude range, speech rate, harmonics-to-noise ratio (HNR), and one-third octave spectral values (as a measure of nasality) can be used to detect sarcasm in audio [23]. Rachel Rakov et al. [24] have developed a model for automatic sarcasm detection. By using K-means clustering, they applied sequential modeling of categorical representations of pitch and intensity contours. They discovered that some particular intensity and pitch contours are indicative of sarcastic speech. Vocal cues were utilized by Rockwell et al. [25] in order to detect sarcasm. Their results indicated that intense low-pitched utterance with slow cadence has higher probability of being sarcastic. Lœvenbruck, Hélène et al. [26] discovered that irrespective of linguistic context, acoustic features can be used to detect sarcasm even in French speech. They also observed that the majority of sarcastic utterances had some common properties like increased f0 modulations and utterance lengthening. Woodland and Voyer [27] showed that prosodic features like intonation and stress play an important role in detecting sarcasm.

Another method to extract audio features is to use Mel-frequency cepstral coefficients (MFCCs). Every sound signal can be uniquely represented as an envelope of time power spectrum, which is also known as Mel-frequency cepstrum (MFC). Human's vocal tract takes a unique shape for unique sounds, and this shape is represented in the time power spectrum of the sound. A MFC is represented by its coefficients which are known as MFCCs.

Tiwari et al. [28] have used MFCCs, as a compact and an effective way, to represent the audio files for a speech processing task.

According to Castro and Santiago et al., multimodal cues can help to improve sarcasm detection [29]. They have observed a reduction in relative error rate of sarcasm when using multimodal cues instead of individual modalities. A hierarchical multimodal architecture has been used by Gu and Yue et al. [30] for classifying sentiment and emotion from text and audio. Cai et al. [31] developed a multimodal

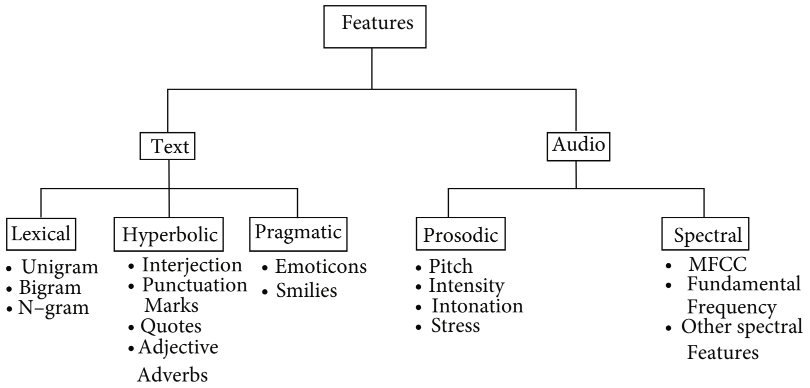

Figure 1: Types of text and audio features used for sarcasm detection.

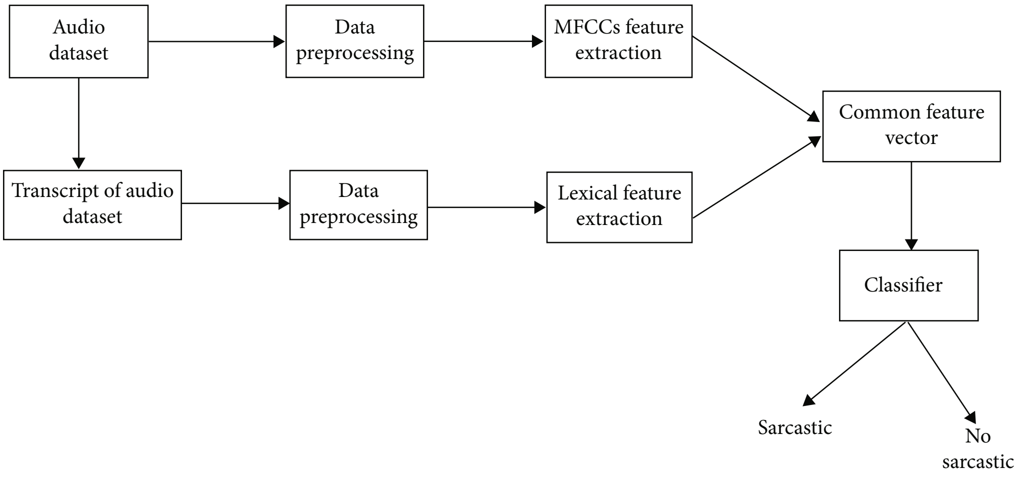

Figure 2: Overall proposed system model for sarcasm detection.

sarcasm detection model to detect sarcasm from tweets. They considered three modalities, text features, image features, and image attributes and constructed a hierarchical fusion model to carry out sarcasm detection.

The types of features used for detecting sarcasm in text and audio are summarized in Figure 1.

3. Methodology

This section deals with the overall methodology of the proposed work. It starts with dataset description, data preprocessing followed by feature extraction algorithm for both text and audio. Finally, extracted features are feed to the classifier to train the proposed model for classification as shown in Figure 2.

3.1. Dataset. In this article, the Mustard dataset was used [20]. The dataset is compiled from popular TV shows including Friends and The Big Bang Theory. It consists of audio-visual utterances, accompanied by its context of the scenario, annotated with its respective sarcasm labels. Textual version of each instance is available in a transcript file which consists of utterance-text, context-text, utterance-speaker, context-speaker, and sarcasm label. The dataset consists of 690 videos, out of them, 345 videos having sarcastic and nonsarcastic labels for every sentences.

3.1.1. Preprocessing. Before to proceed with the dataset, we applied preprocessing on it. In the preprocessing step, the entire clip was segmented into one-second clips. These one-second clips would be classified into sarcastic or nonsarcastic, which would later help in classification tasks of the entire audio clip. A similar approach was used by Pandharipande et al. [23] in order to get enhanced results for their emotion detecting model working on audio files.

Sarcasm label for each second of the clip was manually annotated. For example, if there is eleven-second clip which is sarcastic in its 4th to 6th second, then its output label list would be represented as [0,0,0,1,1,1,0,0,0,0,0], where 1 would represent that the one-second segment is sarcastic and 0 would represent that it is nonsarcastic. The visual representation of the same is shown in Figure 3.

3.2. Feature Extraction. This section elaborate the feature extraction method for both audio as well as textual data.

3.2.1. Audio Data Feature Extraction. Each of the 690 audio files of the dataset was converted from .mp4 extension format to .wav extension format. Further, Mel-frequency cepstral coefficients (MFCCs) was extracted from every .wav file using Python's librosa library. The files were read by the "librosa. load()" function with a sampling rate of 22050 Hz. Each loaded file was converted into its MFCCs representation using

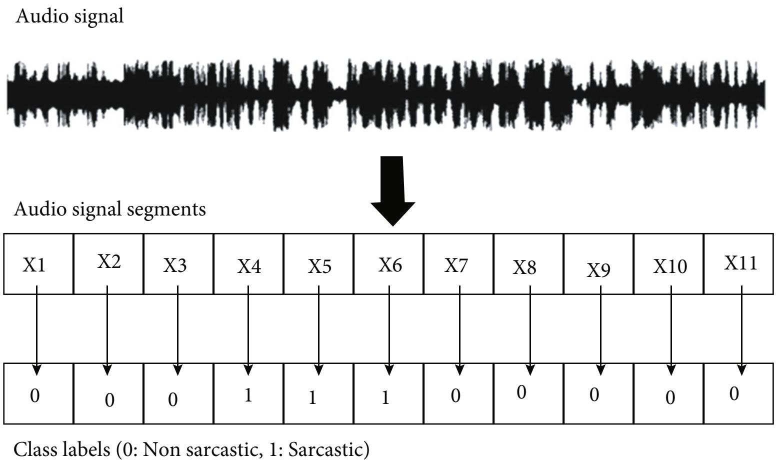

Figure 3: Annotation of every segment of the audio clip.

```
Data:dataset → Audio Utterances from Mustard Dataset
Result:audio_mfcc → mfcc representation of all audio utterances
for audio in dataset do
    audio_wav ← Convert audio from .mp4 to .wav
    a_mfcc ← Convert audio_wav to mfcc format
    append a_mfcc to audio_mfcc
end
```

Algorithm 1: Extracting MFCCs from audio file.

the function "librosa.feature.mfcc()" having parameter values as number of fast fourier transform windows (n_fft) = 2048, hop_length = 512 and number of MFCCs (n_mfcc) were 13. The steps to carry out these operations for the dataset is described in Algorithm 1.

Due to the variable size of audio clips, the MFCC representation of each clip is generated in different lengths. In order to make the input uniform for the training model, the MFCCs were sliced according to the window size of (13, 46), which is equivalent to the size of a one-second audio clip. These one-second segmented audio clips would map one-to-one with the output labels which were created during the preprocessing of dataset. These steps have been shown in Algorithm 2. These MFCC segments along with their output labels are used to train a deep learning model (also known as audio segment model) which detects sarcasm in an audio clip of one-second duration.

This audio segment model consists of a recurrent neural network (RNN). The input layer of this model is a long short-term memory (LSTM) layer that would accept a 46×13 sized matrix and would convert it to a 1-d array of 64 length. The output from this layer would be passed as an input to one more LSTM layer which also generates a 1-d array of 64 length. The next layer is a dense layer having activation function as rectified linear unit (Relu) and a dropout value of 30%. The final layer is a dense layer with a softmax activation function which classifies whether the MFCC segment is sarcastic or not.

3.2.2. Textual Data Feature Extraction. For each audio file present in the dataset, there was a corresponding transcript available, from which utterance-text and sarcasm labels were fetched and stored into a pandas dataframe. The sarcasm label was modified using label encoder where "True" was converted to 1 and "False" to 0. By using the tokenizer available from the Keras preprocessing text library, all the texts were tokenized. Then, "texts_to_sequences" function of the tokenizer was used to vectorize each sentence. Due to varying lengths of texts, the vectors generated are also of different lengths and as a result padding would be required to make the input size uniform. Since the majority of vector length was less than 60, the input length of the model was fixed at 60. And, as a result, those vectors having a size less than 60 were padded.

3.3. Proposed Model. In this section, all the three classification models are described. It starts with classification using audio segment model, where only audio features are used for classification. Further, it describes the classification using text model, where only textual features are used for classification. Finally, it deals with hybrid model for classification, where both audio and text features are combined for classification.

3.3.1. Classification Using Audio Segment Model. The audio segment model works on audio clips of fixed length 1 second, but in the dataset, the audio clips are of varied lengths. So, in order to classify the entire audio clip into sarcastic or not sarcastic, the last layer from the audio segment model is

```
Data: audio_mfcc → mfcc representation of all sarcastic audio utterances
      window_list → window list for sarcastic instances
Result: slice_mfcc → sliced mfcc of 1 sec
      Y → sarcastic or not sarcastic
for i in range(len(audio_mfcc)) do
    s =0
    e =46
    p =0
    while e< len(audio_mfcc[i]) do
        segment ← audio_mfcc[i][s:e]
        append segment to slice_mfcc
        if p >= window_list[i][0] and p <= window_list[i][1] then
          y_seg ← sarcastic
        else
          y_seg ← not sarcastic
          append y_seg to y
          p ← p+1
        end if
    end while
    s = e
    e = e+46
end for
```

Algorithm 2: Slice each MFCC into 1 second segments.

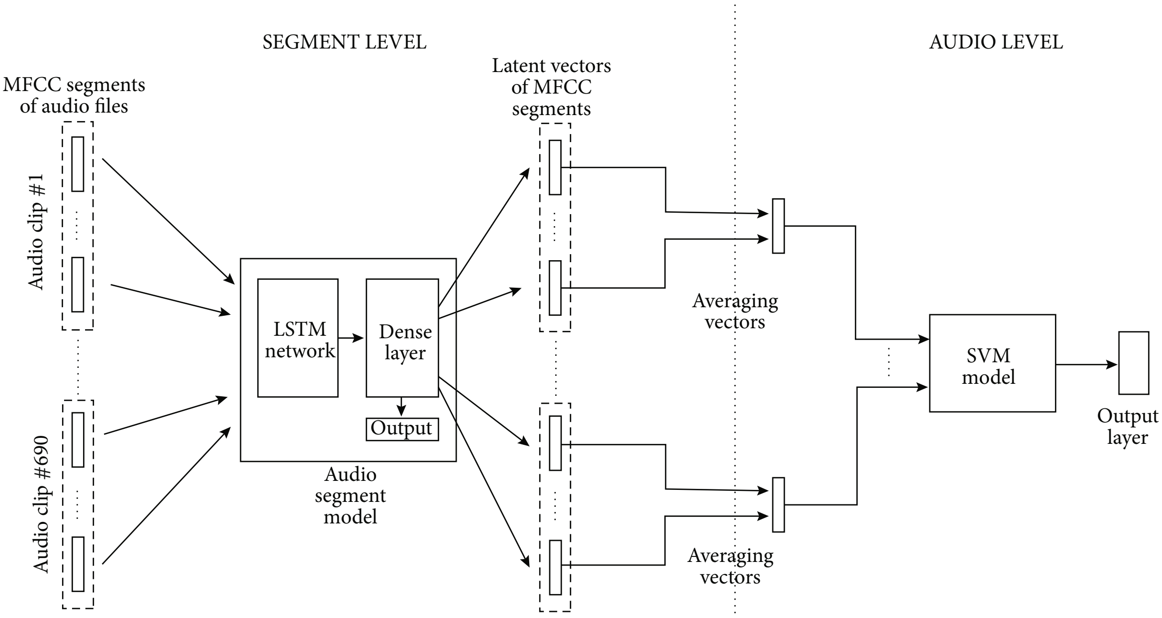

Figure 4: Classification using audio segment model.

removed. After removing the last layer, it would encode the (46, 13) input audio segment to a 64 length array. As the audio segment model works on 1-second segments of audio clips, the output of all segments of a clip would be averaged out to generate a single output for the entire clip. These averaged outputs of all segments for each audio clip would be given as an input to a support vector machine (SVM) model, which would classify the audio clip into either sarcastic or not sarcastic. The visual representation of the model is shown in Figure 4.

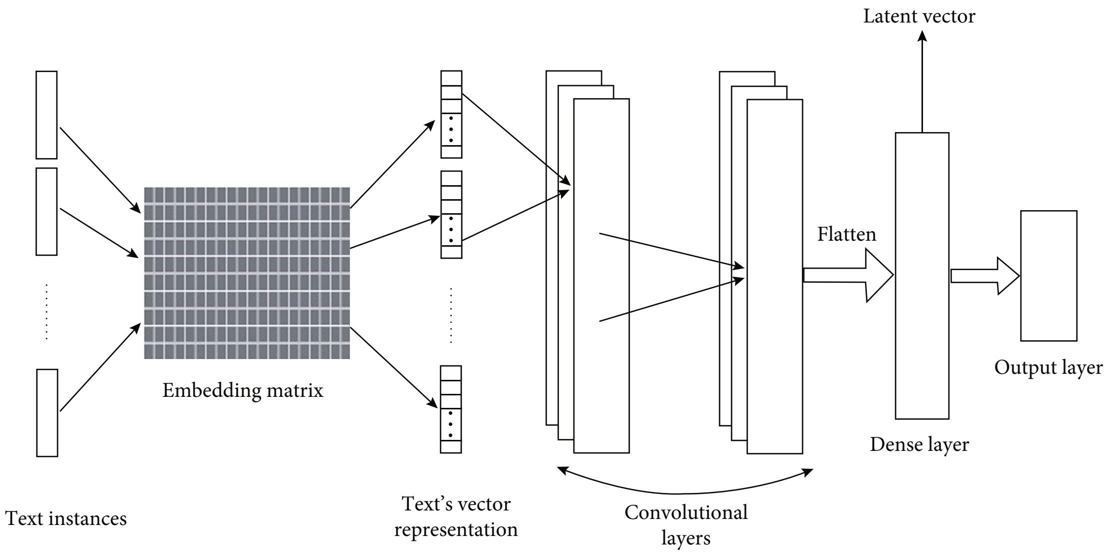

Figure 5: Classification using textual model.

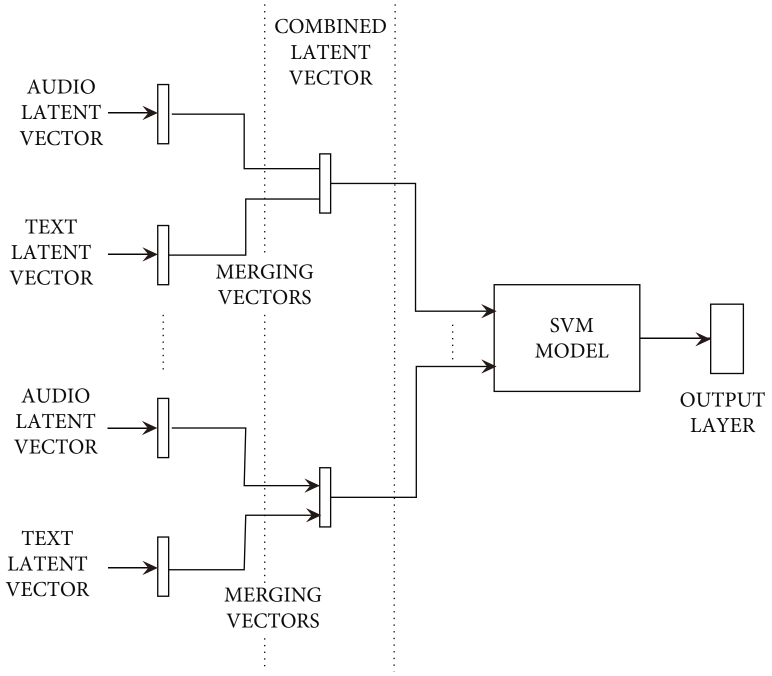

Figure 6: Classification using hybrid model.

Table 1: Comparison of audio model.

Model	Precision	Recall	F1-score
Santiago et al. [28]	65.10	62.60	62.70
Audio model	*73.66*	*66.36*	*69.71*

Table 2: Text model with different word embedding.

Word embedding	Precision	Recall	F1-score
Gensim Word2Vec	*67.5*	*66.66*	*67.08*
BERT	61	61	61
ELMo	54.85	54.34	54.36

Table 3: Hybrid model with different classifiers averaged across 5-folds.

Classifiers	Precision	Recall	F1-score	Accuracy
SVM	*73.26*	65.39	*69.01*	*67.10*
Logistic regression	67.24	*66.49*	66.64	66.38
Gaussian naive Bayes	67.73	62.23	64.77	63.33
Multilayer perceptron	62.45	60.70	61.27	60.72

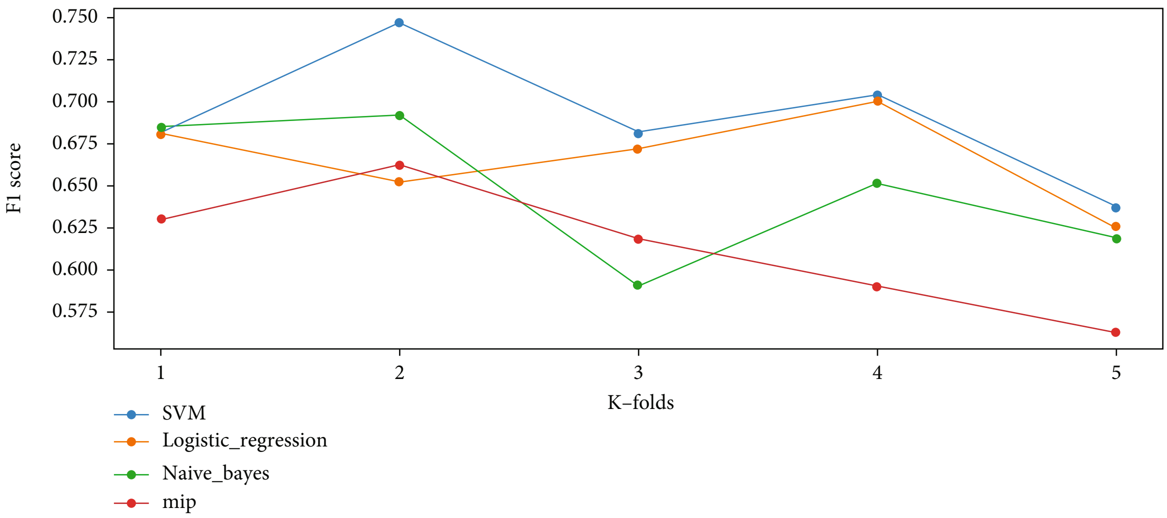

FIGURE 7: F1-scores across the different *K*-folds for various classifiers.

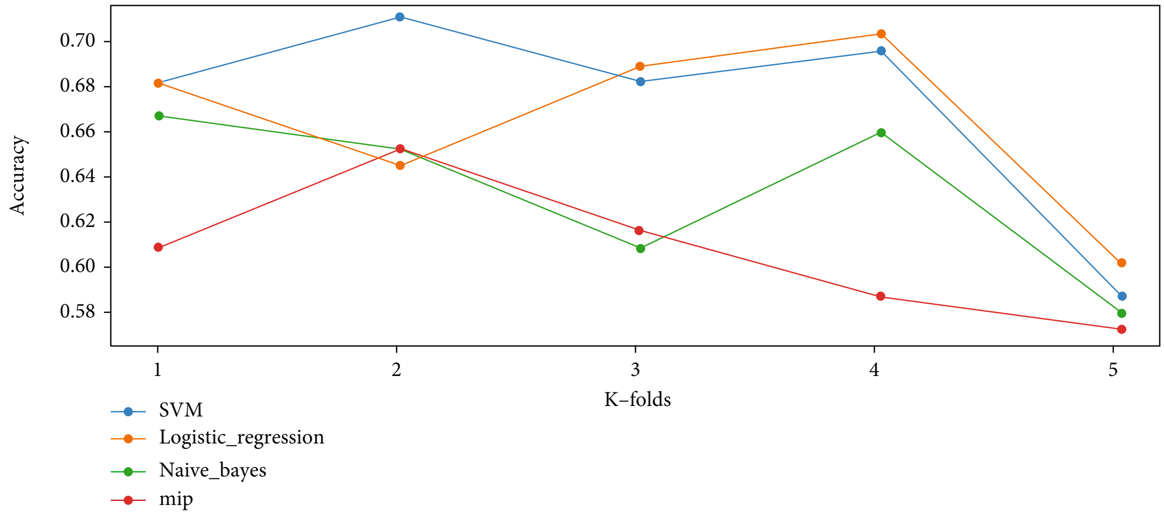

FIGURE 8: Accuracy across different *K*-folds for various classifiers.

3.3.2. Classification Using Textual Model. After each instance of the dataset is preprocessed and padded, it used to train a model which would detect whether the text is sarcastic or not. The first layer of the model is an input layer, which would accept a vector of length 60. This layer would be connected to an embedding layer which would convert the 60 length array to a 300 length array. Instead of randomly assigning weight values to the (60,300) matrix, these weights were fetched from the pretrained model of the genism package. If the word from the vocabulary is present in the genism model, then its vector is fetched otherwise it will be filled with zeros.

The embedding layer is followed by 2 conv1d layers, each of which is followed by a max_pooling layer. Now the output is flatten, and it is followed by a dense layer with Relu activation function. The final layer is also a dense layer with a softmax classifier, which identifies whether the input text is sarcastic or not. The visual representation of the model is shown in Figure 5.

3.3.3. Classification Using Hybrid Model. In order to identify sarcasm by using both text and audio, the last layer from each of the models is removed to convert the models into encoders. The audio encoder model will be used to generate the vector representation of the entire audio file. Similarly, after removing the last layer from the text model, it would encode the text input to a 100 length array.

The output of both the encoder models would be combined to form a 164 length array representation of audio and text features of the clip. Now, this array would be given as input to a support vector machine classifier which would classify it into either sarcastic or nonsarcastic. The visual representation of the model is shown in Figure 6.

4. Results and Discussion

The experimental results of the proposed methodology have been discussed in this section. Firstly, we detected sarcasm by using only audio features of the dataset. In Table 1, the

Table 4: Results of hybrid model with different classifiers.

Modality	Model	Precision	Recall	F1-score
Text (T)	Text model	67.5	*66.66*	67.08
	Santiago et al. [28]	60.9	59.6	59.8
Audio (A)	Audio model	73.66	66.36	69.71
	Santiago et al. [28]	65.1	62.6	62.7
Hybrid (T+A)	Hybrid model	*75.11*	66.28	*70.35*
	Santiago et al. [28]	64.7	62.9	63.1

Table 5: Instances where audio features were more useful than text features (stress was given to the italics part of the text).

Sample of sarcastic instances
Wow, what an *amazing night* this has turned out to be #sarcasm
Oh wow look at the *most realistic doughnuts* in a video game #sarcasm
Wow, that's a huge discount, I'm not buying anything!! #sarcasm
Aha, great night #sarcasm
Wow, publicly promoting "The 1%" concept, great! #sarcasm

Table 6: Instances where text features were more useful than audio features.

Sample of sarcastic instances
I love waiting forever for my doctor #sarcasm
I love the crowed library with no seat. #sarcasm
I love to be ignored #sarcasm
This is the perfect solution for sarcasm detection #not
I never late in the office #joking

results of our audio model are compared with Santiago et al. [28], for 80 : 20 ratio of training and testing data. The attained result of audio model outperformed the existing model with 7% higher F1-score.

Next, only text was used to detect sarcasm. In this approach, deep learning based on three pretrained word embedding model was used in the text model, and its effects were observed. The comparison for these cases are being shown in Table 2.

Finally, both audio and text modalities were used to detect sarcasm. In order to train the hybrid model, the dataset was randomly split into training and testing sets with 80% of the dataset used for training. In order to evaluate the hybrid model, we performed a 5-fold cross validation. This would ensure that our model performs well even with unseen data. Table 3 contains the averaged results of 5-folds for hybrid models with different classifiers.

The F1-score along with different K-folds for all classifiers are shown in Figure 7, and the Accuracy is shown in Figure 8.

Our best models of each modalities are compared with the models proposed by Santiago et al. [28] in Table 4.

In the model where only text is used as an input to detect sarcasm, an F1-score of 67.08 is achieved. Similarly, when only audio is used, we get an F1-score value of 69.71. In the case of the hybrid model, where both audio and text are used as an input to detect sarcasm, an F1-score of 70.35 was observed.

Further, analyzing the instances of the dataset where the hybrid model correctly identifies sarcastic utterance, whereas the text model is unable to do so, it was found that the majority of these text instances, by itself, cannot be considered sarcastic. In these cases, it is the way in which the sentence is spoken which makes it sarcastic. Some of these cases are shown in Table 5.

In the examples shown in Table 5, one can observe that it is not possible to detect sarcasm just from text. By observing the audio of these instances, it was found that it was the stress given by the speakers on the bold part of the sentence which made it sarcastic.

Similarly, opposite scenarios are also possible where sarcasm cannot be detected by using only audio features; in these cases, it is the text itself which makes it sarcastic and not the way of text delivery. Some of these examples are shown in Table 6. In these examples, the sentences are said with a straight face so it would be difficult to detect sarcasm by using audio only. It is the textual features like polarity change and presence of contradicting sentiments which make it sarcastic, for example, considering waitressing a complex and critical activity or needing to listen to snores while sleeping.

5. Conclusion and Future Direction

In this article, we proposed a hybrid model for sarcasm detection in conversational data. In this method, both text and audio features are combined to detect sarcasm. Three models were created, first the text model which works with only textual features, second the audio model which works with only audio features, and third the hybrid model which works with both text and audio features. The text and audio features for the hybrid model were fetched as latent vectors from the other two models, respectively. From the obtained results, one can see that the hybrid model gives the best performance among the three models. This is due to text features and audio features compensating for each other's shortcomings. So, these results support our hypothesis that using both text and audio increases the probability of detecting sarcasm.

In the future direction of the proposed work, it can be explored that the recent advancements in representational learning try to improve our result further. We would like to explore the multilingual embeddings for the efficient code-mixed representations.

Abbreviations

MFCC:	Mel-frequency cepstral coefficients
CNN:	Convolutional neural network
LSTM:	Long short-term memory
AI:	Artificial intelligence
RNN:	Recurrent neural network
RELU:	Rectified linear unit
SVM:	Support vector machine.

Data Availability

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

The authors of this manuscript declared that they do not have any conflict of interest.

Acknowledgments

The authors extend their appreciation to the researchers supporting project number (RSP-2021/384) King Saud University, Riyadh, Saudi Arabia.

References

[1] C. Davenport, “Google Assistant’s duplex will call businesses for you to set up appointments,” https://www.androidpolice.com/2018/05/08/google-assistant-will-call-businesses-set-appointments/.

[2] R. González-Ibánez, S. Muresan, and N. Wacholder, “Identifying sarcasm in twitter: a closer look,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 581–586, Portland, Oregon, 2011.

[3] R. Kreuz and G. Caucci, “Lexical influences on the perception of sarcasm,” in *Proceedings of the Workshop on computational approaches to Figurative Language*, pp. 1–4, Rochester, NY, 2007.

[4] J. Tepperman, D. Traum, and S. Narayanan, “"Yeah right": sarcasm recognition for spoken dialogue systems,” *Ninth International Conference on Spoken Language Processing*, 2006.

[5] E. Savini and C. Caragea, “Intermediate-task transfer learning with BERT for sarcasm detection,” *Mathematics*, vol. 10, no. 5, p. 844, 2022.

[6] T. S. Shekhawat, M. Kumar, U. Rathore, A. Joshi, and J. Patro, “IISERB brains at SemEval 2022 task 6: a deep-learning framework to identify intended sarcasm in English,” https://arxiv.org/abs/2203.02244.

[7] M. Shrivastava and S. Kumar, “A pragmatic and intelligent model for sarcasm detection in social media text,” *Technology in Society*, vol. 64, article 101489, 2021.

[8] Y. Du, T. Li, and M. S. Pathan, “An effective sarcasm detection approach based on sentimental context and individual expression habits,” *Cognitive Computation*, vol. 14, no. 1, pp. 78–90, 2022.

[9] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, “Sarcasm as contrast between a positive sentiment and negative situation,” in *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 704–714, Seattle, Washington, USA, 2013.

[10] R. J. Kreuz and R. M. Roberts, “Two cues for verbal irony: hyperbole and the ironic tone of voice,” *Metaphor and Symbol*, vol. 10, no. 1, pp. 21–31, 1995.

[11] A. Utsumi, “Verbal irony as implicit display of ironic environment: distinguishing ironic utterances from nonirony,” *Journal of Pragmatics*, vol. 32, no. 12, pp. 1777–1806, 2000.

[12] P. Carvalho, L. Sarmento, M. J. Silva, and E. De Oliveira, “Clues for detecting irony in user-generated contents: oh...!! it's" so easy",” *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, 2009, pp. 53–56, Hong Kong, China, 2009.

[13] M. Zhang, Y. Zhang, and F. Guohong, “Tweet sarcasm detection using deep neural network,” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: technical papers*, pp. 2449–2460, Osaka, Japan, 2016.

[14] J. Pennington, R. Socher, and C. D. Manning, “Glove: global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, 2014.

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

[16] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[17] D. Chen and C. D. Manning, “A fast and accurate dependency parser using neural networks,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 740–750, Doha, Qatar, 2014.

[18] I. A. Farha and W. Magdy, “Benchmarking transformer-based language models for Arabic sentiment and sarcasm detection,” in *Proceedings of the sixth Arabic natural language processing workshop*, pp. 21–31, Kyiv, Ukraine, 2021.

[19] Y. Zhang, Y. Liu, Q. Li 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.

[20] D. Faraj and M. Abdullah, “Sarcasmdet at sarcasm detection task 2021 in Arabic using arabert pretrained model,” in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pp. 345–350, Kyiv, Ukraine (Virtual), 2021.

[21] H. Nayel, E. Amer, A. Allam, and H. Abdallah, “Machine learning-based model for sentiment and sarcasm detection,” in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pp. 386–389, Kyiv, Ukraine (Virtual), 2021.

[22] M. Bedi, M. Shivani Kumar, S. Akhtar, and T. Chakraborty, “Multi-modal sarcasm detection and humor classification in code-mixed conversations,” *IEEE Transactions on Affective Computing*, 2021.

[23] R. Rakov and A. Rosenberg, “Sure, I did the right thing: a system for sarcasm detection in speech,” *Interspeech*, , pp. 842–846, Lyon, France, 2013.

[24] P. Rockwell, “Lower, slower, louder: vocal cues of sarcasm,” *Journal of Psycholinguistic Research*, vol. 29, no. 5, pp. 483–495, 2000.

[25] H. Loevenbruck, M. B. Jannet, M. d'Imperio, M. Spini, and M. Champagne-Lavau, “Prosodic cues of sarcastic speech in French: slower, higher, wider,” in *Interspeech 2013-14th Annual Conference of the International Speech Communication Association*, pp. 3537–3541, Lyon, France, 2013.

[26] J. Woodland and D. Voyer, “Context and intonation in the perception of sarcasm,” *Metaphor and Symbol*, vol. 26, no. 3, pp. 227–239, 2011.

[27] V. Tiwari, “MFCC and its applications in speaker recognition,” *International Journal on Emerging Technologies*, vol. 1, no. 1, pp. 19–22, 2010.

[28] S. Castro, D. Hazarika, V. Pérez-Rosas, R. Zimmermann, R. Mihalcea, and S. Poria, “Towards multimodal sarcasm

detection (an obviously perfect paper)," https://arxiv.org/abs/1906.01815.

[29] Y. Gu, K. Yang, S. Fu, S. Chen, X. Li, and I. Marsic, "Multimodal affective analysis using hierarchical attention strategy with word-level alignment," *Proceedings of the Conference. Association for Computational Linguistics. Meeting*, vol. 2018, p. 2225, 2018.

[30] Y. Cai, H. Cai, and X. Wan, "Multi-modal sarcasm detection in twitter with hierarchical fusion model," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2506–2515, Florence, Italy, 2019.

[31] M. A. Pandharipande and S. K. Kopparapu, "Audio segmentation based approach for improved emotion recognition," in *TENCON 2015-2015 IEEE Region 10 Conference*, pp. 1–4, Macao, China, 2015.