JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 39, 909-933 (2023)
DOI: 10.6688/JISE.202307_39(4).0013

B²GRUA: BERTweet Bi-Directional Gated Recurrent Unit with Attention Model for Sarcasm Detection

RAVINDER AHUJA+ AND S. C. SHARMA
Electronics and Computer Discipline
Indian Institute of Technology Roorkee Saharanpur Campus
Saharanpur, 247001 India
E-mail: {ahujaravinder022+; scs60fpt}@gmail.com

Sentiment analysis of social media text containing opinions about the product, event, or service is used in various applications like election results prediction, product endorsement, and many more. Sarcasm is a form of sentiment in which people use positive words to express negative feelings. While communicating verbally, people express sarcasm using hand gestures, facial expressions, and eye movements. These clues are missing in text data, making sarcasm detection challenging. Because of these challenges, scholars are interested in detecting sarcasm in social media texts. The feature extraction technique is an important component in a sarcasm detection model. Most solutions use GloVe, word2vec, or general-purpose pre-trained models for feature extraction. The GloVe/word2vec techniques ignore words that are not present in their vocabulary leading to information loss, require more extensive data for training and generating exact vectors, and ignore contextual information. A general-purpose pre-trained model overcomes the limitations of GloVe/word2vec models but cannot learn features from the social media text due to informal grammar, abbreviations, and irregular vocabulary. In this view, the BERTweet model (trained on social media text) is applied to generate sentence-level semantics and contextual features. The Bi-GRU model processes these features to learn long-distance dependencies from both directions (forward and backward), and the self-attention layer is applied on top of the Bi-GRU model to remove redundant and irrelevant information. This work presents a hybrid method called B²GRUA that combines the strengths of the BERTweet pre-trained model, bi-directional gated recurrent unit and attention mechanism (Bi-GRUAM) for classifying text into sarcastic/non-sarcastic. The efficacy of the proposed model is evaluated on three benchmark datasets, namely SemEval 2018 Task 3.A, iSarcasm, and 2020 shared sarcasm detection task (Twitter data). It is observed from the results that the proposed model outperformed state-of-the-art models on all the datasets (24% better on the iSarcasm dataset and around 2% on both the 2020 shared sarcasm detection task and SemEval 2018 Task 3.A dataset). ANOVA one-way test is applied to validate the results statistically.

Keywords: attention networks, Bi-GRU, figurative languages, pre-trained models, twitter data

1. INTRODUCTION

Numerous social media sites are available in this digital age, including Twitter, Facebook, Instagram, Reddit, and others, through which people from all over the world communicate. People post their opinion/views/sentiments about the event/product/service in messages, text, images, video, or multimodal form [1-3] on these platforms. The sentiment analysis of the posts available on social media platforms is used in various applications like decision-making scenarios [4], product endorsement [5], election result prediction [6],

Received January 4, 2022; revised July 12 & September 2, 2022; accepted September 8, 2022.
Communicated by Meng Chang Chen.
+ Corresponding author.

and many more. Sarcasm is a type of sentiment analysis in which people express a negative feeling using positive words. Due to the presence of sarcastic comments, it becomes difficult to understand the actual opinion or feeling. The definition of sarcasm is "the use of remarks that mean the opposite of what they say, made to hurt someone's feelings or to criticize something humorously." For example, "I'd like to thank Michele Obama for making the fruit snacks in the lunch room 90 % tinier! Really changed my whole life with that one" [7]. People use slang, emojis, bashes, and grammatically incorrect sentences, which makes detecting sarcasm challenging. While communicating verbally, people use facial expressions and hand gestures to represent sarcasm. These clues are not present in the textual data, making it very challenging to detect sarcasm in the text, even for humans. Therefore, an accurate sarcasm detection model is required for text data. Researchers have previously applied machine learning models like naïve bayes, support vector machine, random forest, and logistic regression [8]. These methods require human involvement for feature extraction. Deep learning algorithms improved the performance of sarcasm detection [3, 9]. Deep learning and machine learning algorithms require pre-processing (stemming and lemmatization, stop words removal, removal of digits and punctuation, *etc.*) of data that is time-consuming in large datasets [3, 11]. Nowadays, transformer models like BERT, RoBERTa, and XLNet [11, 12] have performed better than state-of-the-art models in various natural language processing tasks like sentiment analysis, machine translation, and many more. These models are trained on general text from Wikipedia, books, and stories. The nature of the text on Twitter is different due to the limit on the size of the tweet. Social media post contains informal grammar, slang, emoticons, abbreviations, *etc.* Thus, these pre-trained models are not suitable for social media data. Earlier studies have evaluated their model on their datasets in place of benchmark datasets which is not justifiable [11, 13].

This paper proposes a hybrid model combining BERTweet [14] and Bi-GRU with an attention mechanism for sarcasm detection on three benchmark datasets. BERTweet is the pre-trained model for English tweets, and it will generate contextual word embedding. On the contextual embedding, Bi-GRU with attention mechanism is applied to detect sarcasm. We have also done an ablation study to know each component's impact on the performance of the hybrid model. **Through this study, we specifically make the following contributions:** (i) A hybrid model is proposed called B^2GRUA based on contextual word embedding generated through BERTweet and Bi-GRU with an attention mechanism to detect sarcasm on three datasets (iSarcasm, SemEval 2018 Task 3.A, and 2020 shared task sarcasm detection-Twitter dataset only). The hyperparameters of the Bi-GRU model are tuned using grid search methodology; (ii) We have compared the proposed approach with pre-trained models (six models), machine learning (seven models), deep learning (six models), and existing models presented in the literature. The proposed model has given 24% higher performance (F1-score) on the iSarcasm dataset and around 2% higher on both the 2020 shared task sarcasm and 2018 SemEval Task 3.A dataset; (iii) A comparison of results produced by recent studies on the text modality dataset with the multimodal dataset is performed to understand the gap between these two tasks; (iv) An ablation study demonstrates each component's impact on the hybrid model's performance.

The remainder of the paper has the following sections: Section 2 describes deep learning, machine learning, and transformer-based models for sarcasm detection presented in the literature; Section 3 describes datasets used in this study; Section 4 is about the pro-

posed model; Section 5 is about the results and discussion containing experimental setup, baseline models used in this study, pre-processing techniques, experimental results, ablation study, comparison of multimodal with text only dataset, discussion and limitations of this study; Section 6 contains conclusion and future scope.

2. RELATED WORK

Sarcasm can be detected using a linguistic approach and a computational approach. This section presents various computational methods for sarcasm detection used in the past.

2.1 Machine Learning Based

M. Ducret *et al.* [15] have tried linguistic features, and their combination, such as stylistic, text, word complexity, and psychological, to detect sarcasm. They have used LIWC, VAD, and VADER to extract the features from the text. They applied a random forest classifier to the features. They have found that combining linguistic and count features with context have produced better results (f1-score 70%). They have found that with contextual information performance of the models is improved. X. Guo *et al.* [16] have proposed a latent optimization methodology in adversarial neural transfer. Their approach has improved the performance of transfer learning methods by considering different losses (domain-specific and adversarial) to accommodate each other. N. Pawar *et al.* [17] have applied KNN, SVM, and random forest algorithms to the four features extracted from the Twitter dataset. The four features extracted are punctuation-based, pattern-based, syntactic and semantics, and sentiment based. According to their findings, the random forest algorithm outperformed all other models tested. R. Ortega-Bueno *et al.* [18] have introduced a new dataset on irony in three variants of Spanish (Spanish, Mexican, and Spanish news) called IroSvA. The dataset is considered over short messages (tweets and news comments) and annotated by native speakers of each variant. The number of training samples is 2400 and 600 test samples in each dataset (Cuban variant, Mexican, and Cuban). Authors have experimented several machine learning models on three datasets for irony detection.

2.2 Deep Learning Based

S. Oprea *et al.* [19] have introduced the iSarcasm dataset collected from users through surveys. They distinguish between intended and perceived sarcasm. The perceived sarcasm is for audience interpretability, and the intended is corresponding to the author's utterance. They have applied computational methods (3CNN, LSTM, SIARN, MIARN, Dense-LSTM, and Att-LSTM) and manual modeling methods on various sarcasm datasets. They have found that manual labeling performed better on the iSarcasm dataset. A. Kamal *et al.* [20] have proposed a model called CAT-BiGRU, which combines CNN, Bi-GRU, and attention layers for self-deprecating sarcasm detection in Twitter data. They have used GloVe word embedding (200 dimensions) based on Twitter data to convert words to numeric representation. They have considered seven datasets (six standard and one created) in their study. They have considered CNN, LSTM, BiLSTM, CNN-LSTM, and CNN-BiLSTM as base models and compared them with their proposed approach. They have found that adding attention layers improved the performance of the model. D. M. Ashok *et al.* [21] have pro-

posed a model combining Bi-LSTM and CNN to detect sarcasm in Twitter data. Hyperparameters of the Bi-LSTM model are tuned using a genetic algorithm. They have applied BERT for converting text to numeric features. Their approach has performed better than CNN and LSTM-CNN models. They have found that features generated by BERT contributed to the model's performance. J. Lemmens *et al.* [22] have proposed an ensemble model which consists of five components, namely CNN, CNN-LSTM, MLP, SVM, and AdaBoost, for sarcasm detection. They have applied CNN, CNN-LSTM, MLP, and SVM models for detecting sarcasm. The output of these algorithms, along with the length of the context and response, Vader sentiment score of the context is given to AdaBoost classifier (decision tree as base model) to detect sarcasm. R. Xiang *et al.* [23] have introduced a Chinese dataset for irony detection called Ciron. The dataset consists of 8700 posts from the Weibo platform. The dataset was labeled (5 labels) by five Chinese students. Authors have applied LR, SVM, NB, LSTM, CNN, Bi-LSTM, Bi-LSTM-AT, and BERT models to classify the posts into different labels. They observed from the results that the BERT model had performed better among all the models applied. P. Golazizian *et al.* [24] have introduced a new irony dataset in the Persian language. The dataset has 4339 tweets from the telegram channel, OfficialPersianTwitter1, and Twitter, manually labeled by a telegram bot with 12 annotators. Authors have applied the Bi-LSTM model, Bi-LSTM with attention, fastText embedding, without pre-training, and two transfer learning methods to classify the tweets into irony and non-irony. Benamara F *et al.* [25] collected 7,724 tweets in the French language from 2014 to 2016 based on hashtags such as #irony, #sarcasm, and the existence of some keywords such as Holland and Valls. The tweets were annotated by four annotators using text only (without contextual information). Authors have applied three tasks: figurative language identification, non-figurative tweets classification based on polarity, and figurative/non-figurative tweets classification using polarity on their dataset. A. T. Cignarella *et al.* [26] have introduced the IronITA task, which is the detection of irony and different types of irony in Italian tweets. Authors have collected text from Sentipolc corpus, Hate Speech Corpus, LaBuonaScuola corpus, and TWITTIRÒ corpus. The total samples are 4849 tweets, 3977 in the training set, and 872 in the test set. The dataset was annotated at the finer level by four native Italian speakers.

2.3 Transformer Based

K. Pant *et al.* [27] have applied the RoBERTa model to detect sarcasm on the Reddit and Twitter datasets. They have considered three forms of input (response only, response-context, and response context separately) to find the importance of context in the model's performance. They have found that adding contextual information helps improve the model's performance. S. Javdan *et al.* [28] have applied NBSVM, BERT, LCF-BERT, Bi-GRU-CNN, XLNet, Bi-LSTM-CNN, IAN, and BERT-AEN to detect sarcasm on Reddit and Twitter datasets. They have found that LCF-BERT has performed better on the Twitter dataset and BERT-base model with the response-only feature. A. T. Handoyo *et al.* [29] have applied the data augmentation technique to create more samples to balance the dataset. Their approach uses GloVe word embedding and the RoBERTa model on four sarcasm detection datasets. They have found that data augmentation increases the performance of the models. M. Shrivastava *et al.* [12] have proposed a model based on BERT for sarcasm detection. Their method was compared against SVM, LSTM, CNN, LR, Bi-LSTM, and

attention models. They have discussed various standard datasets of sarcasm/irony in their study. They have found that transformer models can improve the performance of the sarcasm detection model. A. Khatri *et al.* [30] have applied SVM, LR, NB, and RF algorithms to the features extracted with GloVe word embeddings and contextual embeddings generated through BERT. They found that logistic regression with GloVe word embedding has the highest performance among all the models, with an F1-score of 0.690. A. Kalaivani. *et al.* [31] have applied five machine learning models, namely SVM, LR, XGBoost, NB, and RF, to the features extracted through TF-IDF and Doc2Vec to detect and identify sarcasm. They have considered context only, the response only, and context with the response from the dataset. They have applied the BERT model also on Twitter and Reddit datasets. They have found that the BERT model has performed better on both datasets among all the models applied. H. Gregory *et al.* [32] have proposed a transformer ensemble model combining BERT, RoBERTa-large, ALBERT, XLNet, and RoBERTa for sarcasm detection on the Twitter dataset. They have also applied LSTM and GRU models considering word embeddings generated through pre-trained transformer models. They have concluded that the ensemble of the transformer model can detect sarcasm efficiently. A. K. Jena *et al.* [33] have proposed a C-Net (Contextual Network) model for sarcasm detection. They have applied SVM, Naïve Bayes, logistic regression, SGD, and XGBoost models from machine learning, Bi-LSTM, RNN, ELMo deep learning, and XLNet, BERT, RoBERTa pre-trained models as the baseline for sarcasm detection. They have found that context information helps in sarcasm detection. D. Faraj *et al.* [34] have applied a voting ensemble model considering two versions (large-arabertv02 and base-arabertv01) of the AraBERT transformer to detect sarcasm in Arabic text. They have considered the dataset of subtask1from WANLP 2021. They compared their model with AraBERTv02, XLM-R, mBERT (cased and uncased) and found that their model outperformed all the models applied. H. Xie *et al.* [35] have proposed a multi-dimensional relation model which considers the relationship between different dimensions such as arousal (excited–calm) and valence (positive-negative) in a deep neural network for dimension score prediction on three-dimensional (valence, arousal, and irony) Chinese dataset. There are two modes of the proposed model, namely, internal and external. The relationship between dimensions is included in the sentence representation in internal mode. In contrast, the linear regression model is applied in the external mode to capture the relationship and refine the predicted score. They have found that their model outperformed other deep learning models and the internal mode outperformed the external mode. A. Agrawal *et al.* [36] have applied XLNet and BERT, two pre-trained models on SemEval 2018 Task 3 dataset. SemEval 2018 Task 3 has two parts, part A is to classify the text into irony and non-irony (binary class), and part B is to classify the text into non-ironic, situational irony, ironic with polarity contrast, and ironic without polarity contrast (multiclass). On the binary classification task XLNet model has performed better than the BERT model, whereas the BERT model has performed better than XLNet on the multiclass classification task. R. A. Potamias *et al.* [11] have proposed a hybrid model, a combination of the RoBERTa pre-trained transformer model and a recurrent convolutional neural network to detect irony and sarcasm on four benchmark datasets. They have compared their proposed model with state-of-the-art and baseline models from machine learning, deep learning, and pre-trained models. They have observed from the results that the proposed model has outperformed all the models. R. Ahuja *et al.* [37] have proposed a pre-trained model called LMTweets, which was trained on 500k tweets and social

media contents to extract the features from datasets. The extracted features are given as input to the CNN model for classification. The pre-trained model designed was based on BERT base architecture. The hybrid model they designed has outperformed all the models on three benchmark datasets: SemEval 2018 Task 3.A, Riloff, and SARC (politics).

Researchers have used various machine, deep, and transformer models for sarcasm detection. However, no one has utilized the transformer model, which is trained on the domain-specific dataset, which can improve the model's performance [38]. We have applied the BERTweet transformer model, which is trained on tweets to convert the text into contextualized word embedding. Further, Bi-GRU with attention mechanism is used to classify the text into sarcastic/non-sarcastic and irony/non-irony classes.

3. DATASET DESCRIPTION

In this study, three standard datasets are considered are as follows:

(i) iSarcasm dataset: This dataset [19] was collected through a survey of Twitter users. The users were asked to give one sarcastic and three non-sarcastic tweets. They have to explain why the tweet is sarcastic and rephrase the tweet, which will provide a non-sarcastic meaning. PA approach was used to collect three annotations to each tweet, and the dominating one was assigned as a label. A total of 1236 responses were received corresponding to the survey conducted. A total of 1236 sarcastic and 3708 non-sarcastic tweets were collected. After applying the quality filter, 777 sarcastic and 3707 non-sarcastic were left out, as given in Table 1. A few examples of sarcastic and non-sarcastic text/tweets are given in Table 2.

Table 1. Information about the dataset considered.

Dataset	Number of Tweets	No. of Sarcastic Tweets	Number of Non-Sarcastic Tweets
iSarcasm [19]	4,484	777	3707
D. Ghosh [41]	6800	3400	3400
SemEval-2018 Task 3.A [10]	4618	1911	2707

Table 2. Sample tweets.

Label	Text
iSarcasm	
Non-Sarcastic	House cleaned. Me cleaned. Ocado order unpacked. Must be time for celebrity cooks
Sarcastic	Because infidelities make everything right.
2020 Sarcasm Detection Shared Task dataset (response only)	
Sarcastic	Such shame that ppl like @USER stand for pakistani actors but dont hv spine to stand
Non-Sarcastic	This photo is particularly great because it was taken in West Yorkshire. It shows a
SemEval-2018 Task 3.A Dataset	
Irony	Hey there! Nice to see you Minnesota/ND Winter Weather
Not Irony	My whole life is just "oh ok".

(ii) 2020 Sarcasm Detection Shared Task: This dataset covers two social media platforms: Twitter and Reddit. We have considered the Twitter dataset only from https://github.

com/marti-duc/Sarcasm-Project/tree/main/Data repository (accessed on 12 February 2021). It consists of 5000 tweets in training and 1800 tweets in the test set. Out of 5000 tweets, 2500 are sarcastic, and 2500 are non-sarcastic tweets. In the test dataset, 900 tweets are sarcastic, and 900 are non-sarcastic. The statistics of the dataset used are given in Table 1. The dataset is balanced, with an equal number of sarcastic and non-sarcastic samples. A few examples of sarcastic and non-sarcastic text/tweets are given in Table 2.

(iii) SemEval 2018 Task 3 Dataset: This dataset contains two classes, ironic and non-ironic, which is collected from Twitter using hashtags such as #irony, #sarcasm, and #not over the period from 01 December 2014 to 04 January 2015 [10]. The data was cleaned by removing non-English tweets, retweets, duplicates, and XML escaped characters replacement. The dataset is publicly available at https://github.com/Cyvhee/SemEval2018-Task3/ (accessed on 14 July 2022). There is a total of 4618 tweets, out of which 3483 samples are in the training set and 784 samples in the test set. The number of non-ironic tweets is 1911, and non-ironic tweets are 2707. A few examples of irony and non-irony tweets are given in Table 2.

4. PROPOSED METHODOLOGY

The proposed model architecture is given in Fig. 1. Firstly, the input is split into the smallest pieces called tokens using a fastBPE tokenizer. The special tokens called [CLS] and [SEP] is added to the tokenized input. [CLS] token is added at the beginning of the sequence and used for classification tasks. [SEP] token is used to separate two sentences; if a single sentence is present, it is added at the end. The tokens generated are mapped to a unique integer (Es) from the embedding table. This input is passed to the BERTweet model. The BERTweet model produces a vector of size 768 (floats) for each token as an output. The output vector corresponding to the [CLS] token is passed to the Bi-GRU model for learning long-distance dependencies in the word sequence and generates a feature vector. The feature vector generated by the Bi-GRU model has redundant and irrelevant informa-

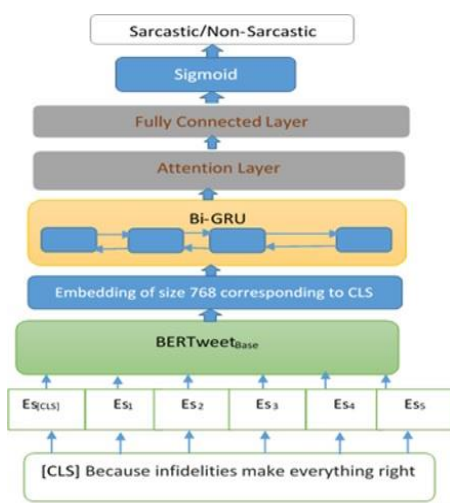

Fig. 1. Proposed model architecture.

tion. A self-attention layer was applied to the feature vector generated by Bi-GRU to capture relevant information. The final layer is a fully connected layer with a sigmoid activation function. The sigmoid activation function's output is the class's probability (sarcastic/irony or non-sarcastic/non-irony). This part of the paper explains the proposed approach, which has two stages: Stages 1 and 2. Stage 1 is to generate contextual word embedding by the social media text trained transformer model (BERTweet), and Stage 2 is to learn the long-distance dependency using sequential models (Bi-GRU). A detailed description of the two stages of the proposed approach is as follows:

Stage 1-Contextual Word Embedding (BERTweet Model): Word embedding represents the words as real value low dimensional vectors. There are mainly two types of embedding techniques, namely, frequency-based and prediction-based. A frequency-based method creates vectors corresponding to text based on the frequency of words in the text (count vectorizer, TF-IDF). These methods cannot capture syntactic, semantic, and contextual information. Prediction-based word embedding creates vectors of the text using neural networks and previous knowledge (GloVe, word2vec). These methods cannot generate vectors using contextual information, require a large corpus for training, and cannot handle out of vocabulary words. For example, Sentence 1: "I left my phone on the left side of the table." In this sentence word "left" will be converted to the same vector by prediction-based methods, although they have different meanings at different places in the sentence. These methods generate closest vectors for opposite words, such as "cold" and "hot," which causes contextual and sentimental loss. On the other hand, contextual word embedding will generate a different vector for the word "left" as its meaning depends upon context. Different contextual word embedding techniques presented in the literature are ELMo, USE, and transformer-based models like BERT and many more. Transformer-based contextual word embeddings have shown better results in various text classification tasks [11, 37, 39, 40].

The BERT is a sequence model designed by Google in 2018. It takes input (text) and generates a contextual representation of it by using encoder architecture taken from the transformer. BERTbase consists of 12 encoder architectures. Every encoder cell is neural network architecture consisting of three processes: multi-head self-attention layer, add & normalize layer, and feedforward, as shown in Fig. 2. The multi-head attention layer extracts the most relevant features from the input. It consists of various matrices operations in series. These extracted features are normalized using add & normalize layer and given to the feedforward neural network. The output of the feedforward neural network is given to the next encoder cell, and this process is repeated for other encoder cells. The input to

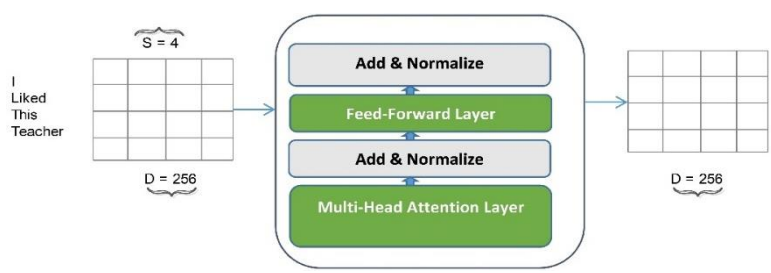

Fig. 2. Encoder cell diagram.

the encoder is a size input_length × embedding_dimension matrix, and the output (produced by the feedforward neural network) is also of the same size input_length × embedding_dimension.

BERT model is trained on text from Wikipedia or books. The transformer-based models trained on a domain-specific dataset can improve the performance of the classification model [37, 41]. As we have considered social media datasets in this study, that's why the BERTweet model is used. BERTweet is a variant of the BERTbase model, which is trained only on social media texts. BERTweet is based on BERTbase architecture and trained using the RoBERTa procedure. BERTweet is trained on 850 M English tweets with 16B word tokens. For training, Tweets from 01/2012 to 08/2019 are downloaded from http://archive.org/details/twitterstream. COVID-19-related tweets (5M) are also considered in the pre-training of BERTweet. Non-English tweets are removed using fastText, and tweets are normalized by converting the URL link to an HTTP URL and user mention to @USER. TweetTokenizer from the NLTK library is used to tokenize the sentence. The tweets which are retweeted and tweets with lengths less than ten and more than 64 are filtered out. The parameters used in BERTweet are: (i) maximum sequence length is set to 128; (ii) Adam optimizer is used with learning rate 0.0004 with a batch size of 7k using 8V100 GPUs; (iii) number of epochs are 40; (iv) 12 layers; (v) Hidden layer size:768; (vi) Number of attention heads: 12. The last hidden state of the last encoder model is contextual word embedding of the word, which is of size 768. The past studies [12, 27, 29, 31-33, 36] have utilized transformer models to classify text into sarcastic/non-sarcastic. They have not used the capability of pre-trained models by adding other neural networks to the output produced by them [11]. The previous studies [11, 28, 30] have utilized the transformer model, which is trained on conventional text from books, Wikipedia, stories, and news to generate contextual word embedding. These models are capable of learning features from social media text; hence social media trained model (BERTweet) is applied to generate the contextual features.

Stage 2-Bi-Directional GRU with Attention Mechanism: In the previous stage, contextual word embedding is extracted using the BERTweet model is passes through the Bi-Directional GRU layer. LSTM and GRU are the variants of the RNN model which solve the vanishing/exploding gradient problem of RNN. These models have excellent capabilities for learning long-term dependencies. LSTM and GRU models are suitable for sequential information modeling tasks such as text classification. GRU is a less complex variant of the LSTM model. GRU has fewer parameters, fewer data to generalize, and less training execution time than LSTM. The structure of GRU is given in Fig. 3. It consists of two gates: **Reset Gate (r_t):** It is used to determine how much old information can be ignored, and **Update Gate (z_t):** It is used to determine how much past information must be carried down to the next state. The output (h_t) of the GRU cell is calculated using the current input (x_t) and previous state (h_(t-1)) with the supervision of two gates (reset and update). The gates and GRU cell output are given in Eqs. (1)-(4).

$$reset\_gate(r_t) = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (1)$$

$$update\_gate(z_t) = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (2)$$

$$\tilde{h}_t = tanh(W_h x_t + U_z(r_t \odot h_{t-1}) + b_h) \quad (3)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (4)$$

W_r, U_r, U_z, U_h are the weight matrices, b_r, b_z, b_h are the bias vectors, σ is a sigmoid function, and $\odot$ is a product operator.

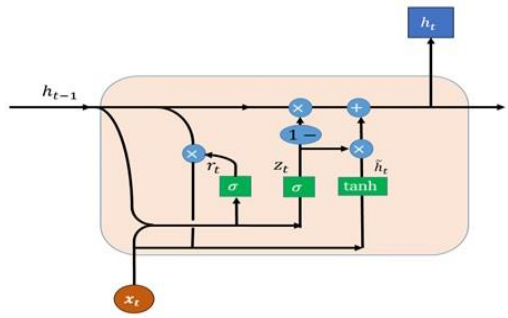

Fig. 3. Structure of GRU cell.

The Bi-GRU model has two GRU units, one in the forward direction and the other in the backward direction. Bi-GRU can learn information from previous and subsequent data while considering current input. This model encapsulates the information in the form of feature vectors. The Bi-GRU model is defined as follows:

$$\overrightarrow{h_t} = GRU_{Foreward}(x_t, \overrightarrow{h_{t-1}}), \tag{5}$$

$$\overleftarrow{h_t} = GRU_{Backward}(x_t, \overleftarrow{h_{t+1}}), \tag{6}$$

$$h_t = \overleftarrow{h_t} \oplus \overleftarrow{h_t}. \tag{7}$$

$\overrightarrow{h_t}$ is the forward GRU state, $\overleftarrow{h_t}$ is the backward GRU state, and h_t is the concatenation of the forward and backward state.

The studies presented in the literature [11, 37] showed that adding a neural network model (CNN/LSTM) to the features generated by pre-trained models improved the performance of the sarcasm detection model. The CNN model cannot capture long-range dependencies, and the LSTM model cannot learn long-range dependencies from both directions (forward and backward). Previous studies [9, 20, 42] showed that Bi-LSTM/Bi-GRU model performed better on the sarcasm detection task. To this end, the Bi-GRU model is applied to the features generated by the BERTweet model. Bi-GRU is not capable of capturing important information, and it generates redundant and irrelevant features. The self-attention layer is used after the Bi-GRU model to capture important information and remove redundant and irrelevant features. The self-attention layer relates different positions of a single sequence to compute a representation of the same sequence. This enables us to learn the correlation between the current words and the previous part of the sentence. For example, in the sentence "The monkey did not eat the bananas because it was too full," Here word 'it' refers to the monkey.

If in place of full, it is much, then the word 'it' refers to bananas. The attention mechanism helps understand whether the word 'it' is related to monkeys or bananas. Authors [20, 43, 44] of previous related work showed that adding an attention layer can improve the model's performance. So, we have applied scaled dot-product self-attention on the top

of the Bi-GRU model. The hyperparameters of the Bi-GRU model are optimized with the grid search technique and given in Table 3. The algorithm of the proposed model can be seen from Algorithm 1.

Algorithm 1 B^{2}GRUA: BERTweet-BiGRU with Attention Mechanism Algorithm

Procedure: BERTweet + BiGRU with Attention Mechanism
Input: Tweets/Text
Output: Sarcastic or Non-Sarcastic
1. for every text t_i from input, do
   Tokenized words (W_{tokens}) ← Tokenization of t_i
   Word Representation (size 768) ← BERTweet (Tokenized words)
2. Bi-GRU(L) ← Word Representation for [*CLS*] token
3. DropoutLayer (*DL*) ← Bi-GRU(L)
4. Attention Layer (*AL*) ← DropoutLayer (*DL*)
5. Dense Layer (Sigmoid) ← Attention Layer (*AL*)
6. Probabilities (P_1 and P_2) ← Dense Layer (Sigmoid)
7. **if** $P_1 \geq 0.5$, **then**
   Sarcastic/Irony
   else
   Non-Sarcastic/Non-Irony

Where t_i represents text, L represents the Bi-GRU layer, DL represents the dropout layer, *AL* represents the attention layer, P_1 and P_2 are probabilities of sarcasm and non-sarcastic, respectively.

Table 3. Hyperparameters used in the Bi-GRU model.

Hyperparameter	Value
Number of layers	2
Hidden Units	128
Learning rate	1e-3
Optimizer	Adam
Dropout	0.5
Number of epochs	40

5. EXPERIMENTAL RESULTS AND DISCUSSION

This section consists of the experimental setup, baseline models, pre-processing techniques used in baseline models, evaluation measures, experimental results, comparative study with baseline models and SOA models, ablation study, comparison of multimodal with text only dataset, discussion, and limitations.

5.1 Experimental Setup

The experiments were performed on the Google Colab Pro-environment, which provides a shared GPU (K80/T4/V100) with 24 GB RAM. The proposed model, transformer models, and deep learning models are implemented using a simple transformer (http://si-

mpletransformers.ai/docs/classification-specifics/) library, Keras (https://keras.io/) framework with TensorFlow (https://www.tensorflow.org/)) as backend is used to implement deep learning models. The machine learning models are implemented using Python's scikit-learn [45] library. Accuracy, recall, precision, auc-roc, f1-score, and balanced accuracy parameters are employed for performance measurement.

5.2 Baseline Models

We have applied seven machine learning, six deep learning, and six transformer models as baseline models as mentioned below:

(A) Machine Learning Algorithms: We have implemented seven machine learning models, namely, (i) NB; (ii) KNN; (iii) DT; (iv) RF; (v) SVM; (vi) LR; and (vii) GB, because these have been used as baseline models in various sarcasm detection studies. The TF-IDF features with unigram, bigram, and trigram to represent text numerically with machine learning models.

(B) Deep Learning Algorithms: We have implemented six deep learning models, namely, (i) CNN; (ii) LSTM; (iii) GRU; (iv) GRU-Pooling; (v) Bi-LSTM-AM; (vi) Bi-GRU-AM as baseline models from the deep learning category. The GloVe word embedding is used to convert text into numeric representations of 300 dimensions.

(C) Pre-Trained Models: Pre-trained models like BERT and RoBERTa have produced better results in various NLP tasks such as text classification, text generation, question answering, text summarization, and many more [46-48]. We have implemented six pre-trained models: BERT, DistilLBERT, RoBERTa, Electra, XLNet, and BERTweet. BERTweet is trained on the tweets only, which makes it more suitable for this study because we have considered the social media data.

(D) Pre-Processing Methods: Social media text contains noise and lots of undesired information like acronyms, slang, *etc*. Due to this unwanted information, the performance of the models is affected. For cleaning the data, we have applied the following steps: (i) conversion to lower case; (ii) Stop words are removed; (iii) Digits removal; (iv) Stemming; (v) emoji and emoticons removal; (vi) URL removal; (vii) User mentions are removed.

The framework used with baseline models and the proposed approach is given in Fig. 4. The dataset is pre-processed (for machine learning and deep learning models only) using the techniques mentioned in Section 5.2.4. After pre-processing, the TF-IDF features (unigram, bi-gram, and trigram) are extracted for machine learning models, and GloVe (300 dimensions) embedding is used to generate features for deep learning models. Seven machine learning models (given in Section 5.2.1) are applied on the TF-IDF features. On Glove word embeddings six deep learning models (given in Section 5.2.2) are applied. Six pre-trained models (given in Section 5.2.3) are applied and evaluated. The performance of state-of-the-art models and baseline models is compared with the proposed model on the basis of accuracy, precision, recall, f1-score, and AUC-ROC. Finally, the results are statistically validated using the One-way ANOVA test.

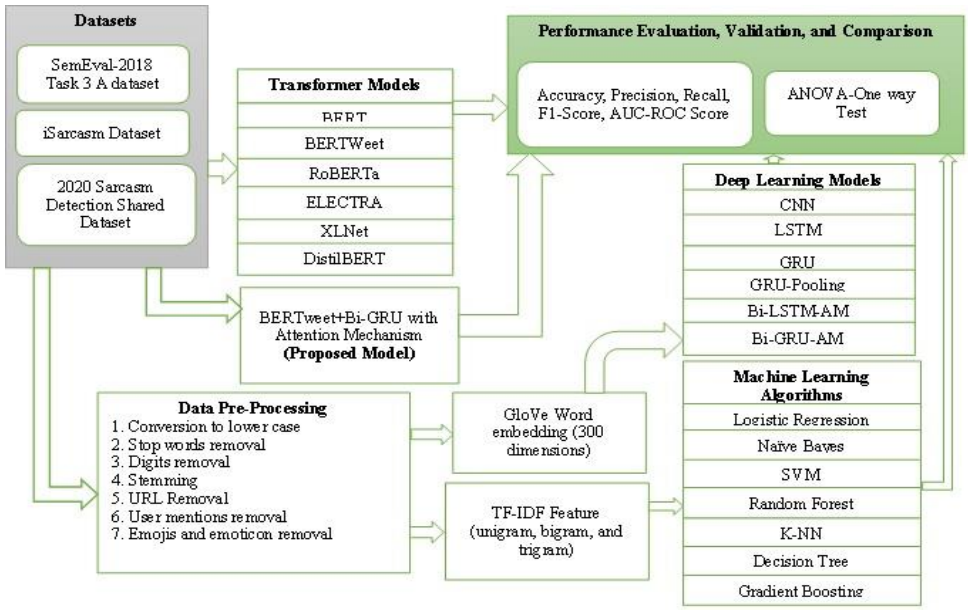

Fig. 4. Framework for baseline models and proposed approach.

5.3 Experimental Results on iSarcasm Dataset

The results of machine learning, deep learning, transformer models, and the proposed approach to classify the text into sarcastic and non-sarcastic on the iSarcasm dataset are given in Table 4. Among all the machine learning models applied, the random forest has performed better with an f1-score value of 0.7626. SVM algorithm has performed almost equally to the random forest with an f1-score value of 0.7586. The DT algorithm reports the worst performance with an f1-score value of 0.7432. Out of all deep learning algorithms, Bi-GRU with the attention model has performed better with an f1-score value of 0.7895. CNN algorithm has performed poorly from deep learning models with an f1-score value of 0.7687. BERTweet has performed better than all pre-trained models with an f1-score value of 0.8360. The Electra pre-trained model from pre-trained models has performed poorly with the f1-score value of 0.7613. The proposed hybrid model has outperformed all the machine learning, deep learning, pre-trained models, and models presented in the literature with an F1-score value of 0.8514.

Table 4. Performance comparison of baseline models, SOA models from literature, and proposed model on iSarcasm dataset.

S. No.	Models	Balanced Accuracy	Precision	Recall	F1-score	AUC
ML Algorithms						
1	GB	0.5244±0.0199	0.7385±0.0199	0.7902±0.0199	0.7580±0.0222	0.5497±0.0313
2	NB	0.5117±0.0183	0.7307±0.0183	0.7891±0.0183	0.7518±0.0179	0.5562±0.0269
3	SVM	0.5224±0.0176	0.7380±0.0176	0.7954±0.0176	0.7586±0.0205	0.5434±0.0275
4	DT	0.5283±0.0226	0.7384±0.0226	0.7515±0.0226	0.7432±0.0236	0.5221±0.0301

5	KNN	0.4997±0.0064	0.7134±0.0064	0.8213±0.0064	0.7513±0.0238	0.4969±0.0334
6	LR	0.4995±0.0010	0.6880±0.0010	0.8286±0.0010	0.7517±0.0211	0.5594±0.0291
7	RF	0.5154±0.0188	0.7497±0.0188	0.8197±0.0188	0.7626±0.0217	0.5583±0.0302
DL Algorithms						
8	CNN	0.5387±0.0031	0.7687±0.0031	0.7687±0.0031	0.7687±0.0031	0.5387±0.0031
9	LSTM	0.5583±0.0231	0.7783±0.0231	0.7783±0.0231	0.7783±0.0231	0.5583±0.0231
10	GRU	0.5675±0.0072	0.7724±0.0164	0.8304±0.0005	0.7736±0.0048	0.5675±0.0072
11	GRU-Pooling	0.5774±0.0052	0.7925±0.0072	0.8325±0.0010	0.7866±0.0045	0.5774±0.0052
12	Bi-LSTM-AM	0.5720±0.0137	0.7811±0.0175	0.8268±0.0095	0.7804±0.0056	0.5720±0.0137
13	Bi-GRU-AM	0.5900±0.0033	0.7908±0.0164	0.8224±0.0112	0.7895±0.0034	0.5900±0.0033
Pre-Trained Models						
14	BERT	0.5639±0.0160	0.7912±0.0155	0.8302±0.0104	0.7957±0.0084	0.5639±0.0160
15	DistilBERT	0.5371±0.0244	0.7695±0.0366	0.8320±0.0051	0.7826±0.0129	0.5371±0.0244
16	Electra	0.5000±0.0001	0.6989±0.0002	0.8360±0.0001	0.7613±0.0003	0.5000±0.0002
17	RoBERTa	0.6648±0.0200	0.8213±0.0051	0.8284±0.0043	0.8239±0.0019	0.6648±0.0200
18	XLNet	0.5134±0.0079	0.7668±0.0427	0.8353±0.0035	0.7707±0.0055	0.5134±0.0079
19	BERTweet	0.6794±0.0095	0.8357±0.0044	0.8557±0.0030	0.8360±0.0053	0.6794±0.0095
Comparison with SOA Models from the Literature						
20	X. Guo *et al.* [16]	–	0.236	0.793	0.4642	–
21	S. Oprea *et al.* [19]	–	0.550	0.701	0.6160	–
22	A. T. Handoyo *et al.* [29]	–	–	–	0.4044	–
23	Our Work	0.7050±0.0121	0.8481±0.0050	0.8569±0.0052	0.8514±0.0048	0.7050±0.0121

5.4 Experimental Results on 2020 Sarcasm Shared Task Twitter Dataset

The results of machine learning, deep learning, transformer models, and the proposed approach to classify the text into sarcastic and non-sarcastic on the 2020 Sarcasm Shared Task dataset are given in Table 5. Among all the machine learning models applied, SVM has performed better with an f1-score value of 0.6650. Next to SVM is a logistic regression algorithm with an f1-score value of 0.6374. The DT algorithm has given worst performance with an f1-score value of 0.5934. The GRU-Pooling model has performed better out of all deep learning algorithms, with an f1-score value of 0.7868. CNN algorithm has performed poorly from deep learning models with an f1-score value of 0.6435. Out of all pre-trained models, BERTweet has performed better with an f1-score value of 0.7554. RoBERTa model has performed poorly from pre-trained models with the f1-score value of 0.6784. The proposed hybrid model has outperformed all the machine learning, deep learning, pre-trained models, and models presented in the literature with an f1-score value of 0.7908.

Table 5. Performance comparison of baseline models, SOA models from literature, and proposed model on 2020 Shared Sarcasm Dataset.

S. No.	Models	Accuracy	Precision	Recall	F1-score	AUC
ML Algorithms						
1	GB	0.5971±0.0741	0.5971±0.0741	0.5971±0.0741	0.5971±0.0741	0.5971±0.0741
2	NB	0.6159±0.1068	0.6159±0.1068	0.6159±0.1068	0.6159±0.1068	0.6159±0.1068
3	SVM	0.6650±0.0540	0.6650±0.0540	0.6650±0.0540	0.6650±0.0540	0.6650±0.0540
4	DT	0.5912±0.0517	0.5949±0.0517	0.6018±0.0517	0.5934±0.0525	0.5932±0.0492

5	KNN	0.5984±0.0405	0.5984±0.0405	0.5984±0.0405	0.5984±0.0405	0.5984±0.0405
6	LR	0.6374±0.0783	0.6374±0.0783	0.6374±0.0783	0.6374±0.0783	0.6374±0.0783
7	RF	0.6238±0.0637	0.6238±0.0637	0.6226±0.0637	0.6260±0.0665	0.6246±0.0680
DL Algorithms						
8	CNN	0.6434±0.0060	0.6435±0.0060	0.6435±0.0060	0.6435±0.0060	0.6900±0.0136
9	LSTM	0.6746±0.0060	0.6727±0.0050	0.6727±0.0050	0.6727±0.0050	0.7390±0.0052
10	GRU	0.6628±0.0046	0.6645±0.0047	0.6645±0.0047	0.6645±0.0047	0.7196±0.0048
11	GRU-Pooling	0.7752±0.0055	0.7913±0.0103	0.8726±0.0202	0.7868±0.0034	0.7752±0.0055
12	Bi-LSTM-AM	0.7447±0.0031	0.7558±0.0132	0.8975±0.0443	0.7650±0.0020	0.7453±0.0031
13	Bi-GRU-AM	0.7495±0.0034	0.7660±0.0089	0.8685±0.0470	0.7622±0.0075	0.7498±0.0033
Pre-Trained Models						
14	BERT	0.7147±0.0073	0.7283±0.0113	0.6855±0.0173	0.7061±0.0087	0.7147±0.0073
15	DistilBERT	0.7069±0.0013	0.7246±0.0131	0.6692±0.0276	0.6952±0.0091	0.7069±0.0013
16	Electra	0.7172±0.0073	0.7309±0.0136	0.6892±0.0349	0.7086±0.0144	0.7172±0.0073
17	RoBERTa	0.6900±0.0038	0.7053±0.0139	0.6547±0.0321	0.6784±0.0114	0.6900±0.0038
18	XLNet	0.6943±0.0069	0.7098±0.0094	0.6580±0.0274	0.6825±0.0128	0.6943±0.0069
19	BERTweet	0.7565±0.0064	0.7748±0.0206	0.7389±0.0209	0.7554±0.0124	0.7565±0.0064
Comparison with SOA Models from the Literature						
20	M. Ducret *et al.* [15]	0.660	0.800	0.669	0.700	
21	J. Lemmens *et al.* [22]	–	0.741	0.746	0.740	–
22	K. Pant *et al.* [27]	–	0.772	0.772	0.772	–
23	S. Javdan *et al.* [28]	–	–	–	0.730	–
24	A. Khatri [30]	–	–	–	0.690	–
25	A. Kalaivani *et al.* [31]	–	0.722	0.722	0.722	–
26	H. Gregory *et al.* [32]	–	0.758	0.767	0.756	–
27	A. K. Jena *et al.* [33]	–	–	–	0.750	–
28	Our Work	0.7835±0.0035	0.7391±0.0106	0.7844±0.0173	0.7908±0.0039	0.7835±0.0035

5.5 Experimental Results on SemEval-2018 Task 3.A Dataset

The results of machine learning, deep learning, transformer models, and the proposed approach on SemEval 2018 Task 3.A dataset is given in Table 6. Among all the machine learning models applied, the BernoulliNB has performed better with an f1-score value of 0.6184. SVM algorithm has performed almost equally to the BernoulliNB with an f1-score value of 0.6063. The KNN algorithm has given worst performance with an f1-score value of 0.3928. The GRU-Pooling model has performed better out of all deep learning algorithms, with an f1-score value of 0.7118. CNN algorithm has performed poorly from deep learning models with an f1-score value of 0.5340. Out of all pre-trained models, DistilBERT has performed poorly, with an f1-score value of 0.5628. The BERTweet pre-trained model from pre-trained models has performed better with the f1-score value of 0.8708. The proposed hybrid model has outperformed all the machine learning, deep learning, pre-trained models, and models presented in the literature with an f1-score value of 0.9045. Some of the misclassified examples by the proposed model are presented in Table 7. These samples do not have any features (punctuation, linguistic, and pattern-based) related to

sarcasm. These are simple statements without any context. Only the author knows why these are marked as sarcastic. Our model failed to detect sarcasm when there is no context was given or no feature (punctuation, linguistic, and pattern-based) was present in the text.

Table 6. Performance comparison of baseline models, SOA models from literature, and proposed model on SemEval-2018 Task 3.A dataset.

S. No.	Models	Accuracy	Precision	Recall	F1-score	AUC
ML Algorithms						
1	GB	0.6298±0.0593	0.6694±0.0593	0.4647±0.0593	0.5389±0.0593	0.6509±0.0702
2	NB	0.6473±0.0379	0.6457±0.0379	0.6016±0.0379	0.6184±0.0379	0.7050±0.0456
3	SVM	0.6312±0.0291	0.6252±0.0291	0.5996±0.0291	0.6063±0.0388	0.6865±0.0480
4	DT	0.6007±0.0381	0.5831±0.0381	0.5822±0.0381	0.5796±0.0390	0.5950±0.0484
5	KNN	0.5379±0.0230	0.5358±0.0230	0.3130±0.0230	0.3928±0.0283	0.5527±0.0175
6	LR	0.6473±0.0390	0.6650±0.0390	0.5540±0.0390	0.5961±0.0558	0.7072±0.0505
7	RF	0.6434±0.0395	0.6453±0.0395	0.5808±0.0395	0.5996±0.0460	0.6913±0.0630
DL Algorithms						
8	CNN	0.6260±0.0085	0.5269±0.0126	0.5595±0.0822	0.5340±0.0371	0.6698±0.0154
9	LSTM	0.6418±0.0151	0.5472±0.0255	0.6294±0.0647	0.5750±0.0187	0.6697±0.0069
10	GRU	0.6917±0.0136	0.7164±0.0064	0.6897±0.0095	0.6907±0.0098	0.6917±0.0136
11	GRU-Pooling	0.7192±0.0072	0.7390±0.0094	0.7093±0.0055	0.7118±0.0051	0.7192±0.0072
12	Bi-LSTM-AM	0.6899±0.0078	0.7150±0.0127	0.6897±0.0075	0.6882±0.0069	0.6817±0.0138
13	Bi-GRU-AM	0.6902±0.0171	0.6937±0.0196	0.6902±0.0171	0.6888±0.0202	0.6792±0.0203
Pre-Trained Models						
14	BERT	0.6428±0.0115	0.5418±0.0140	0.6552±0.0468	0.5922±0.0178	0.6449±0.0118
15	DistilBERT	0.6382±0.0057	0.5403±0.0048	0.5884±0.0384	0.5628±0.0190	0.6297±0.0109
16	Electra	0.7066±0.0181	0.5997±0.0219	0.7891±0.0155	0.6811±0.0136	0.7207±0.0148
17	RoBERTa	0.7397±0.0094	0.6273±0.0110	0.8488±0.0170	0.7213±0.0086	0.7585±0.0086
18	XLNet	0.6724±0.0171	0.5589±0.0146	0.8270±0.0210	0.6670±0.0167	0.6989±0.0173
19	BERTweet	0.8692±0.0061	0.7862±0.0079	0.9759±0.0066	0.8708±0.0050	0.8877±0.0050
Comparison with SOA Models from the Literature						
20	A. Agrawal *et al.* [36]	0.70	0.79	0.70	0.74	–
21	M. Shrivastava, *et al.* [12]	0.69	0.59	0.86	0.69	–
22	R. Potamias A *et al.* [11]	0.82	0.81	0.80	0.80	0.89
23	R. Ahuja *et al.* [37]	0.883	0.865	0.908	0.885	0.898
24	Our Work	0.9051±0.0061	0.8846±0.0065	0.9254±0.0062	0.9045±0.0042	0.9135±0.0038

Table 7. Misclassified examples.

Text	Actual Label	Predicted Label
need about 54 hours of sleep	1	0
because infidelities make everything right.	1	0
i only like 2d women	1	0
just.... wow @hullkrofficial	1	0
need about 54 hours of sleep	1	0

5.6 Comparison of Text Modality with Multimodal Sarcasm Detection Approaches

People can express their views about a particular product or service through different modalities such as text, audio, video, and image. Multimodal sarcasm detection is to know the feelings of the people considering data in different modalities as a whole. S. Castro *et al.* [49] have designed a multimodal dataset called MUStARD which consists of audiovisual utterances with sarcastic/non-sarcastic labels. Their dataset has three modalities: text, audio, and video. They showed that using a multimodal dataset could reduce the error rate by 12.9% compared to using a text-only modality. The authors found that text samples are not able to represent sarcasm, for this, they need some cues from other modalities (audio and/or video). S. K. Bharti *et al.* [50] have applied hybrid deep learning models to detect sarcasm from the multimodal dataset (audio and text). They combined textual and audio features to detect sarcasm in conversational data. They have explored their model on the text-only dataset, audio-only dataset, and multimodal dataset. Their proposed model has given an f1-score value of 70.35 on the multimodal dataset and an f1-score value of 67.08 on text only dataset. The results are 2.27% better on the multimodal dataset as compared to the text-only dataset. The authors have found that only text is not able to reveal the information about sarcasm, it is only the way how that sentence was spoken that makes it sarcastic. N. Ding *et al.* [51] have proposed a multi-level fusion model with residual connection to detect sarcasm in the multimodal dataset (text+audio+video). They have applied their model to single modality (text, audio, and video) and multimodal data (text+audio, audio+video, text+video, and text+audio+video). They have observed that text modality cannot represent sarcasm accurately, and visual and audio modalities are more expressive. Their proposed model showed performance improvement by 4.85% and error rate reduction by 11.8% on the multi-model dataset compared to the text-only dataset. M. U. Salur *et al.* [52] have proposed a soft voting ensemble model to predict sentiment in two multimodel datasets (image and text), namely, MVSA-Single and MVSA-multiple. They have applied the proposed model to both the datasets considering individual modality (text and image) and multimodality (text+image). Their model achieved an f1-score of 66.16% on text only and 72.44% f1-score on text+image in the MVSA-single dataset. Their model achieved an f1-score of 65.19% on text modality and a 71.79 % f1-score on multimodality in the MVSA-multiple datasets. There is an improvement of around 6% in the f1-score on both datasets. S. Sangwan *et al.* [53] have proposed a deep learning model based on a recurrent neural network to detect sarcasm in the multimodal dataset (text+image+transcript). They have considered two multimodal datasets: the gold standard and the silver one. Their proposed model has achieved an accuracy of 66.17% on text modality and 71.5% on the gold standard dataset. Their proposed model has achieved an accuracy of 80.42% on text modality and 84.22% on the silver standard dataset. There is an improvement of 5% accuracy in the gold standard dataset and around 4% in the silver standard dataset compared to the text modality dataset. These studies presented by researchers in the past showed that the performance of computational models for sarcasm detection tasks on multi-model datasets is better as compared to text modality datasets. The text modality is not able to express sarcasm effectively. Other modalities like images, audio, and video can add cues in the textual data to express sarcasm in a better way.

5.7 Ablation Study

Ablation study means to remove the part of the model and understand its effect on the

performance. In the proposed model, there are two components, namely BERTweet and Bi-GRU, with the attention model. We have removed the BERTweet part and evaluated the performance of Bi-GRU with the attention model on GloVe word embedding. Then we removed Bi-GRUAM and evaluated the performance of the BERTweet model only. The results of the ablation study on three datasets are presented in Table 8. It is observed from the results that both components are important in the proposed model. If we have considered BERTweeet, only the f1-score is reduced by 2% on the iSarcasm dataset. If we have considered the Bi-GRUAM model only, then the f1-score is degraded by 7% on the iSarcasm dataset. In the 2020 shared sarcasm detection task, if we consider BERTweet, only the f1-score is reduced by 3%, and if only the Bi-GRUAM model is considered, then f1-score is reduced by 3%. In the case of SemEval 2018 Task 3.A dataset, if we consider BERTweet, only the f1-score is reduced by around 3%, and if only the Bi-GRUAM model is considered, then f1-score is reduced substantially.

Table 8. Ablation study.

Models	Accuracy	Precision	Recall	F1-score	AUC
On Dataset 1 (iSarcasm dataset)					
BERTWeet+Bi-GRU with Attention	0.7050±0.0121	0.8481±0.0050	0.8569±0.0052	0.8514±0.0048	0.7050±0.0121
BERTweet	0.6794±0.0095	0.8357±0.0044	0.8557±0.0030	0.8360±0.0053	0.6794±0.0095
Bi-GRU with Attention	0.5900±0.0033	0.7908±0.0164	0.8224±0.0112	0.7895±0.0034	0.5900±0.0033
On Dataset 2 (2020 Shared Sarcasm Task Dataset)					
BERTWeet+Bi-GRU with Attention	0.7835±0.0035	0.7391±0.0106	0.7844±0.0173	0.7908±0.0039	0.7835±0.0035
BERTweet	0.7535±0.0035	0.7391±0.0106	0.7844±0.0173	0.7608±0.0039	0.7535±0.0035
Bi-GRU with Attention	0.7495±0.0034	0.7660±0.0089	0.8685±0.0470	0.7622±0.0075	0.7498±0.0033
On Dataset 3 (SemEval-2018 Task 3.A Dataset)					
BERTWeet+Bi-GRU with Attention	0.9051±0.0061	0.8846±0.0065	0.9254±0.0062	0.9045±0.0042	0.9135±0.0038
BERTweet	0.8692±0.0061	0.7862±0.0079	0.9759±0.0066	0.8708±0.0050	0.8877±0.0050
Bi-GRU with Attention	0.6902±0.0171	0.6937±0.0196	0.6902±0.0171	0.6888±0.0202	0.6792±0.0203

The ANOVA Statistical test: ANOVA stands for analysis of variance. It is a statistical test used to check whether the means of two or more samples are significantly different or not. ANOVA test is of two types, namely, one-way and two-way. One-way ANOVA is used to determine how one factor affects the response variable. Two-way ANOVA is used to determine how two factors affect the response variable. In our study, only one factor is present: the type of method (machine learning, deep learning, pre-trained models, and proposed approach), and the response variable is performance (f1-score). So, one-way ANOVA was applied in this study. The null hypothesis is that the mean of different samples is equal, and the alternative hypothesis is that there is a significant difference in the mean of different samples. We have used the statsmodels library to implement an ANOVA-one-way test to validate the results statistically. ANOVA test gives p-value and F-statistics as an output. The null hypothesis is rejected if the p-value is less than the significance value (0.05 taken in this study) [54]. The p-value obtained considering the f1-score is 0.000452, which is less than 0.05. Hence, the null hypothesis is rejected, and the means of the f1-score is

different from the proposed approach, machine learning, pre-trained, and deep learning models.

Discussion:

• We have proposed a hybrid model called B^2GRUA, which has two components, BERTweet and Bi-GRUAM. The proposed model outperformed the models [16, 19, 29] on iSarcasm dataset with f1-score value 0.8514 (24% better), models [15, 22, 27, 28, 30-33] on 2020 shared sarcasm dataset with f-score value 0.7908 (2% better), and models [11, 12, 36, 37] on SemEval 2018 Task 3.A dataset with f-score value 0.9037 (around 2% better). The authors [12, 27, 29, 31-33, 36] have applied the BERT/XLNet/RoBERTa/ALBERT pre-trained model for the classification of the sentence into sarcastic/non-sarcastic. They have not utilized the capability of pre-trained models by adding other neural networks to them [11]. These pre-trained models are trained on conventional text from books corpus, stories, and Wikipedia corpus; hence, these models are not suitable for social media text due to usages of irregular grammar, emoticons, typographical errors, abbreviations, *etc.* The authors [11, 28] have applied the RoBERTa/BERT pertained model with other deep learning models such as LSTM and AEN for sarcasm detection. The limitation of their study is that they have not applied a domain-specific pre-trained model which can improve performance. They have applied unidirectional LSTM, which cannot capture the dependency between word sequences from both directions. They have not utilized the capabilities of the self-attention mechanism, which can improve classification performance [43, 55]. R. Ahuja *et al.* [37] have developed a domain-specific pre-trained model (considering Tweets) called LMTweets and applied the CNN model on top of it to classify the text. Their pre-trained model was trained on less data (500,000 tweets) than BERTweet, which is trained on 850M tweets, which limits its prediction capability. BERTweet models training approach is based on the RoBERTa model, which gives robust performance [14], whereas the LMTweets model pre-training is based on the BERT model. They have applied the CNN model, which cannot learn long-distance dependency from sequential data. The attention mechanism which can improve performance is not utilized in their studies. A.Khatri [30] *et al.* have extracted features using BERT and GloVe models from the text. The GloVe model cannot capture contextual information from the text, which helps in detecting the sarcasm. GloVe model is also not able to handle out of vocabulary words which cause information loss. They have applied machine learning models instead of deep learning models, which can give better results [9, 37]. S. Oprea *et al.* [19] applied a manual labeling method to detect sarcasm. The limitation of manual labeling in sarcasm detection is it cannot understand the authors' actual intention. Manual labeling is a time-consuming and expensive process (it requires a lot of language/domain-specific human resources). M. Ducret *et al.* [15] used only linguistic features with machine learning models to detect sarcasm. The authors have not considered other types of features, such as pattern-based, punctuation-based, linguistic, sentiment-based, and syntactic features, which also represent sarcasm. Machine learning models are not a good fit for sarcasm detection tasks because they cannot understand the context of a given text. X. Guo *et al.* [16] used a transfer learning approach using adversarial neural transfer to detect sarcasm. Their model was trained on a smaller dataset which limits its generalization capability. J. Lemmens *et al.* [22] have developed an ensemble model that considers machine learning and deep learning. They have used the GloVe model for word representation which cannot handle out of

vocabulary words and cannot learn contextual information. They have considered the stylometric and emotion characteristics features of machine learning models. Although sarcasm can be represented with other features also, such as hyperbolic features, syntactic features, pragmatics features, punctuation-based features, linguistic features, self-deprecating features, and Twitter-specific features. They have considered CNN and LSTM models, which cannot learn long-distance dependencies in word sequence from both directions. The proposed model combines the strengths of BERTweet (pre-trained model on social media text), Bi-GRU model, and attention mechanism to design an efficient sarcasm detection model. BERTweet is used to convert text into contextualized rich word embedding and sentence semantics, which is given as an input to the Bi-GRU with an attention model for classification. BERTweet is used because it is pre-trained on social media text only, and we have considered the dataset from social media in this study. GRU is used because it takes fewer parameters and takes less training and execution time than the LSTM model. Bi-GRU is a bidirectional GRU that has two GRU layers. One GRU layer is used to capture information in the forward direction, and another is used in the backward direction. The attention layer helps improve the model's performance by focusing on the important words from a sarcasm detection point of view by removing redundant and irrelevant information.

- B^2GRUA (our approach) has produced 24% higher results than the reference study on the iSarcasm dataset and 2% higher on the SemEval 2018 Task 3.A dataset and 2020 sarcasm detection shared task dataset. The difference in the performance of the proposed model on these datasets is because the nature of the iSarcasm dataset is the intended type, and the 2020 shared sarcasm detection dataset is of perceived type (ii) 2020 shared sarcasm dataset contains conversation data (contextual information) which is helpful in sarcasm detection while in iSarcasm dataset contextual information is not present. Our proposed model contains the BERTweet component, which is used to generate contextual word embeddings that majorly impact the performance of the iSarcasm dataset as compared to the 2020 shared sarcasm dataset.

- The ablation study showed that both the components (BERTweet and BI-GRUAM) contribute to the proposed model's performance.

- A comparative study of results produced by recent studies on text-only datasets and the multimodal dataset is performed. It is found that results are better on multimodal datasets than text-only datasets. The difference in the performance is because other modalities like image, audio, and video add clues to the text.

- The proposed approach does not require pre-processing of the datasets, which is a time taking process.

Limitations:

- Only English tweets are considered in this study, although the sarcastic text is available in other languages as well such as Arabic, Hindi, Dutch, Spanish, Hindi-English mixed, and many more.

• A single modal dataset that consists of text is considered in this study. Multimodal datasets having text, images, videos, and audio not considered, which improves the performance of the model.

6. CONCLUSIONS

In this study, we have proposed a hybrid model which consists of contextual word embedding generated by BERTweet and Bi-GRU with an attention model to detect sarcasm. The performance of the proposed model is evaluated on three sarcasm datasets, namely iSarcasm, SemEval 2018 Task 3.A and 2020 shared task on sarcasm detection (Twitter data only). As baseline models, we have taken seven algorithms from machine learning, six from transformers, and six from deep learning. The proposed model has given 24% higher performance than state-of-the-art models on the iSarcasm dataset and around 2% better results on SemEval 2018 Task 3.A and 2020 shared sarcasm detection task. It is observed from the ablation study that contextual word embedding plays an important role in sarcasm detection. In the future, multimodal and multi-lingual datasets can be considered. Other features like linguistic and punctuation-based can be combined with contextual embedding to obtain a rich set of features that may produce better results.

List of Abbreviations

AM: Attention Mechanism	BERT: Bi-Directional Encoder Representation from Transformer
Bi-LSTM: Bi-Directional LSTM	Bi-GRUAM: Bi-directional gated recurrent unit with an attention mechanism
CNN: Convolutional Neural Network	DL: Deep Learning
DT: Decision Tree	ELMo: Embedding from Language Models
GB: Gradient Boosting	GloVe: Global Vectors for Word Representation
GRU: Gated Recurrent Unit	iSarcasm: Intended Sarcasm
KNN: K-Nearest Neighbor	LIWC: Linguistic Inquiry and Word Count
LSTM: Long Short Term Memory	LR: Logistic Regression
MIARN: Multi-dimensional Intra-Attention Recurrent. Network	ML: Machine Learning
MLP: Multi-layer Perceptron	MUStARD: Multimodal Sarcasm Detection Dataset
NB: Naïve Bayes	RF: Random Forest
RoBERTa: Robustly Optimized BERT	RNN: Recurrent Neural Network
SOA: State-of-the-art	SVM: Support Vector Machine
VAD: Valence Arousal Dominance	

REFERENCES

1. A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: A survey," *ACM Computing Surveys*, Vol. 50, 2017, pp. 1-22.

2. N. Malave and S. N. Dhage, "Sarcasm detection on twitter: User behavior approach," in *Intelligent Systems, Technologies and Applications*, S. M. Thampi, L. Trajkovic, S. Mitra, P. Nagabhushan, *et al*., (eds.), Springer, Singapore, 2020, pp. 65-76.
3. A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, M. Abdel-Basset *et al.*, "Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network," *IEEE Access*, Vol. 7, 2019, pp. 23319-23328.
4. S. K. Chandar, H. Punjabi, M. K. Sharda, and J. Murugadhas, "A novel data science approach for business and decision making for prediction of stock market movement using twitter data and news sentiments," *Data Science and Data Analytics: Opportunities and Challenges*, 2021, p. 305.
5. G. Abeza, N. O'Reilly, B. Séguin, and O. Nzindukiyimana, "The world's highest-paid athletes, product endorsement, and twitter," *Sport, Business and Management*, Vol. –, 2017, pp. 332-355.
6. J. Ramteke, S. Shah, D. Godhia, and A. Shaikh, "Election result prediction using twitter sentiment analysis," in *Proceedings of IEEE International Conference on Inventive Computation Technologies*, Vol. 1, 2016, pp. 1-5.
7. S. Mukherjee and P. K. Bala, "Sarcasm detection in microblogs using naive bayes and fuzzy clustering," *Technology in Society*, Vol. 48, 2017, pp. 19-27.
8. S. M. Sarsam, H. Al-Samarraie, A. I. Alzahrani, and B. Wright, "Sarcasm detection using machine learning algorithms in twitter: A systematic review," *International Journal of Market Research*, Vol. 62, 2020, pp. 578-598.
9. A. Kumar and G. Garg, "Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets," *Journal of Ambient Intelligence and Humanized Computing*, 2019, pp. 1-16.
10. C. van Hee, E. Lefever, and V. Hoste, "Semeval-2018 task 3: Irony detection in English tweets," in *Proceedings of the 12th International Workshop on Semantic Evaluation*, 2018, pp. 39-50.
11. R. A. Potamias, G. Siolas, and A.-G. Stafylopatis, "A transformer-based approach to irony and sarcasm detection," *Neural Computing and Applications*, Vol. 32, 2020, pp. 17309-17320.
12. M. Shrivastava and S. Kumar, "A pragmatic and intelligent model for sarcasm detection in social media text," *Technology in Society*, Vol. 64, 2021, p. 101489.
13. J. Ling and R. Klinger, "An empirical, quantitative analysis of the differences between sarcasm and irony," in *Proceedings of European Semantic Web Conference*, 2016, pp. 203-216.
14. D. Q. Nguyen, T. Vu, and A. T. Nguyen, "Bertweet: A pre-trained language model for english tweets," *arXiv Preprint*, 2020, arXiv:2005.10200.
15. M. Ducret, L. Kruse, C. Martinez, A. Feldman, and J. Peng, "You don't say linguistic features in sarcasm detection – ceur-ws.org," http://ceur-ws.org/Vol-2769/paper 83.pdf, 2020.
16. X. Guo, B. Li, H. Yu, and C. Miao, "Latent-optimized adversarial neural transfer for sarcasm detection," *arXiv Preprint*, 2021, arXiv:2104.09261.
17. N. Pawar and S. Bhingarkar, "Machine learning based sarcasm detection on twitter data," in *Proceedings of IEEE 5th International Conference on Communication and Electronics Systems*, 2020, pp. 957-961.

18. R. Ortega-Bueno, F. Rangel, D. Hernández Farıas, P. Rosso, M. Montes-y Gómez, and J. E. Medina Pagola, "Overview of the task on irony detection in Spanish variants," in *Proceedings of Iberian Languages Evaluation Forum, Co-located with 34th Conference of the Spanish Society for Natural Language Processing*, Vol. 2421, 2019, pp. 229-256.
19. S. Oprea and W. Magdy, "isarcasm: A dataset of intended sarcasm," *arXiv Preprint*, 2019, arXiv:1911.03123.
20. A. Kamal and M. Abulaish, "Cat-bigru: Convolution and attention with bi-directional gated recurrent unit for self-deprecating sarcasm detection," *Cognitive Computation*, Vol. 14, 2022, pp. 91-109.
21. D. M. Ashok, A. N. Ghanshyam, S. S. Salim, D. B. Mazahir, and B. S. Thakare, "Sarcasm detection using genetic optimization on lstm with cnn," in *Proceedings of IEEE International Conference for Emerging Technology*, 2020, pp. 1-4.
22. J. Lemmens, B. Burtenshaw, E. Lotfi, I. Markov, and W. Daelemans, "Sarcasm detection using an ensemble approach," in *Proceedings of the 2nd Workshop on Figurative Language Processing*, 2020, pp. 264-269.
23. R. Xiang, X. Gao, Y. Long, A. Li, E. Chersoni, Q. Lu, and C.-R. Huang, "Ciron: a new benchmark dataset for Chinese irony detection," in *Proceedings of the 12th Language Resources and Evaluation Conference of European Language Resources Association*, 2020, pp. 5714-5720.
24. P. Golazizian, B. Sabeti, S. A. A. Asli, Z. Majdabadi, O. Momenzadeh, and R. Fahmi, "Irony detection in Persian language: A transfer learning approach using emoji prediction," in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 2839-2845.
25. F. Benamara, C. Grouin, J. Karoui, V. Moriceau, and I. Robba, "Analyse dópinion et langage figuratif dans des tweets: Présentation" https://hal.archives-ouvertes.fr/hal-01912785/document, 2021.
26. A. T. Cignarella, S. Frenda, V. Basile, C. Bosco, V. Patti, P. Rosso *et al.*, "Overview of the evalita 2018 task on irony detection in Italian tweets (ironita)," in *Proceedings of the 6th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, Vol. 2263. CEUR-WS, 2018, pp. 1-6.
27. K. Pant and T. Dadu, "Sarcasm detection using context separators in online discourse," *arXiv Preprint*, 2020, arXiv:2006.00850.
28. S. Javdan, B. Minaei-Bidgoli *et al.*, "Applying transformers and aspect-based sentiment analysis approaches on sarcasm detection," in *Proceedings of the 2nd Workshop on Figurative Language Processing*, 2020, pp. 67-71.
29. A. T. Handoyo, D. Suhartono *et al.*, "Sarcasm detection in twitter–performance impact while using data augmentation: Word embeddings," *arXiv Preprint*, 2021, arXiv:2108.09924.
30. A. Khatri *et al.*, "Sarcasm detection in tweets with Bert and glove embeddings," *arXiv Preprint*, 2020, arXiv:2006.11512.
31. A. Kalaivani and D. Thenmozhi, "Sarcasm identification and detection in conversion context using bert," in *Proceedings of the 2nd Workshop on Figurative Language Processing*, 2020, pp. 72-76.

32. H. Gregory, S. Li, P. Mohammadi, N. Tarn, R. Draelos, and C. Rudin, "A transformer approach to contextual sarcasm detection in twitter," in *Proceedings of the 2nd Workshop on Figurative Language Processing*, 2020, pp. 270-275.
33. A. K. Jena, A. Sinha, and R. Agarwal, "C-net: Contextual network for sarcasm detection," in *Proceedings of the 2nd Workshop on Figurative Language Processing*, 2020, pp. 61-66.
34. D. Faraj and M. Abdullah, "Sarcasmdet at sarcasm detection task 2021 in Arabic using Arabert pretrained model," in *Proceedings of the 6th Arabic Natural Language Processing Workshop*, 2021, pp. 345-350.
35. H. Xie, W. Lin, S. Lin, J. Wang, and L.-C. Yu, "Multi-dimensional relation model for dimensional sentiment analysis," *Information Sciences*, Vol. 579, 2021, pp. 832-844.
36. A. Agrawal, A. K. Jha, A. Jaiswal, and V. Kumar, "Irony detection using transformers," in *Proceedings of IEEE International Conference on Computing and Data Science*, 2020, pp. 165-168.
37. R. Ahuja and S. C. Sharma, "Transformer-based word embedding with cnn model to detect sarcasm and irony," *Arabian Journal for Science and Engineering*, Vol. 47, 2022, pp. 9379-9392.
38. S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning-based text classification: a comprehensive review," *ACM Computing Surveys*, Vol. 54, 2021, pp. 1-40.
39. C. Colón-Ruiz and I. Segura-Bedmar, "Comparing deep learning architectures for sentiment analysis on drug reviews," *Journal of Biomedical Informatics*, Vol. 110, 2020, p. 103539.
40. H. Sakhrani, S. Parekh, and P. Ratadiya, "Contextualized embedding based approaches for social media-specific sentiment analysis," in *Proceedings of IEEE International Conference on Data Mining Workshops*, 2021, pp. 186-193.
41. Z. Zheng, X.-Z. Lu, K.-Y. Chen, Y.-C. Zhou, and J.-R. Lin, "Pretrained domain-specific language model for natural language processing tasks in the AEC domain," *Computers in Industry*, Vol. 142, 2022, p. 103733.
42. C. Jia and H. Zan, "Context-based sarcasm detection model in Chinese social media using bert and bi-gru models," *CEUR Workshop Proceedings*, Vol. 3150, 2022.
43. R. Pandey, A. Kumar, J. P. Singh, and S. Tripathi, "Hybrid attention-based long short-term memory network for sarcasm identification," *Applied Soft Computing*, Vol. 106, 2021, p. 107348.
44. A. Alishahi, G. Chrupała, and T. Linzen, "Analyzing and interpreting neural networks for NLP: A report on the first BlackboxNLP workshop," *Natural Language Engineering*, Vol. 25, 2019, pp. 543-557.
45. J. Hao and T. K. Ho, "Machine learning made easy: a review of scikit-learn package in python programming language," *Journal of Educational and Behavioral Statistics*, Vol. 44, 2019, pp. 348-361.
46. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv Preprint*, 2018, arXiv:1810. 04805.
47. V. Maslej-Krešňáková, M. Sarnovský, P. Butka, and K. Machoá, "Comparison of deep learning models and various text pre-processing techniques for the toxic comments classification," *Applied Sciences*, Vol. 10, 2020, p. 8631.

48. A. E. Yüksel, Y. A. Türkmen, A. Özgür, and B. Altınel, "Turkish tweet classification with transformer encoder," in *Proceedings of International Conference on Recent Advances in Natural Language Processing*, 2019, pp. 1380-1387.
49. S. Castro, D. Hazarika, V. Pérez-Rosas, R. Zimmermann, R. Mihalcea, and S. Poria, "Towards multimodal sarcasm detection (an obviously perfect paper)," *arXiv Preprint*, 2019, arXiv:1906.01815.
50. S. K. Bharti, R. K. Gupta, P. K. Shukla, W. A. Hatamleh, H. Tarazi, and S. J. Nuagah, "Multimodal sarcasm detection: A deep learning approach," *Wireless Communications and Mobile Computing*, Vol. 2022, 2022, No. 1653696.
51. N. Ding, S.-W. Tian, and L. Yu, "A multimodal fusion method for sarcasm detection based on late fusion," *Multimedia Tools and Applications*, Vol. 81, 2022, pp. 8597-8616.
52. M. U. Salur and İ. Aydın, "A soft voting ensemble learning-based approach for multimodal sentiment analysis," *Neural Computing and Applications*, Vol. 34, 2022, pp. 18391-18406.
53. S. Sangwan, M. S. Akhtar, P. Behera, and A. Ekbal, "I didn't mean what I wrote! Exploring multimodality for sarcasm detection," in *Proceedings of IEEE International Joint Conference on Neural Networks*, 2020, pp. 1-8.
54. H.-Y. Kim, "Analysis of variance (anova) comparing means of more than two groups," *Restorative Dentistry & Endodontics*, Vol. 39, 2014, pp. 74-77.
55. G. Letarte, F. Paradis, P. Giguère, and F. Laviolette, "Importance of self-attention for sentiment analysis," *ACL Anthology*, https://aclanthology.org/W18-5429/, 2021.

Ravinder Ahuja received the B.Tech. in Computer Science and Engineering from Kurukshetra University Kurukshetra Haryana and M.Tech (CSE) from IIT Roorkee, Uttarakhand, India. He is currently pursuing Ph.D. in computer science discipline from IIT Roorkee, Uttarakhand, India. His research interests include text analytics and educational mining.

S. C. Sharma received his Ph.D. degree in Electronics and Computer Science Engineering from IIT Roorkee in 1991. He is currently working as professor at Indian Institute of Technology Roorkee. His research interests include wireless sensor networks, cloud computing, data science, and internet of things (IOT).