Multimedia Tools and Applications (2022) 81:43229–43252
https://doi.org/10.1007/s11042-022-12930-z

Sarcasm detection using deep learning and ensemble learning

Priya Goel[1] · Rachna Jain[2] · Anand Nayyar[3,4] · Shruti Singhal[1] · Muskan Srivastava[1]

Received: 19 May 2021 / Revised: 26 January 2022 / Accepted: 10 March 2022 /
Published online: 21 May 2022

Abstract
Across the globe, there is a noticeable upward trend of incorporating sarcasm in everyday life. This trend can be easily attributed to the frequent use of sarcasm in everyday life, but more specifically to social media and the Internet. This study aims to bridge the gap between human and machine intelligence to recognize and understand sarcastic behavior and patterns. The research is based on using various neural techniques, namely Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Baseline Convolutional Neural Networks (CNN) in an ensemble model to detect sarcasm on the internet. In order to improve the precision of the proposed model, the required dataset is also prepared on different previously trained word-embedding models like fastText, Word2Vec, and GloVe, etc., and their accuracies are compared. The aim is to be able to quantify the overall sentiment of the writer as positive or negative / sarcastic or non-sarcastic to ensure that the correct message is received to the intended audience. The final study revealed that the proposed ensemble model with word embeddings outperformed the other state-of-the-art models and deep learning models considered in this study with an accuracy of around

✉ Anand Nayyar
anandnayyar@duytan.edu.vn

Priya Goel
priyagoel99@gmail.com

Rachna Jain
rachnajain@bpitindia.com

Shruti Singhal
shruti.singhal.2608@gmail.com

Muskan Srivastava
muskan.srivastava1904@gmail.com

[1] Computer Science Department, Bharati Vidyapeeth's College of Engineering, New Delhi, India

[2] Information Technology Department, Bhagwan Parshuram Institute of Technology, New Delhi, India

[3] Graduate School, Duy Tan University, Da Nang 550000, Viet Nam

[4] Faculty of Information Technology, Duy Tan University, Da Nang 550000, Viet Nam

96% for News Headlines dataset, 73% for Reddit dataset, and amongst our proposed ensemble models, Weighted Average Ensemble gave the highest accuracy of around 99% and 82% for both the datasets respectively. Ensemble model used in our study improvised the stability, precision and predictive power of the proposed model.

Keywords Sarcasm · Natural language processing (NLP) · Long short-term memory (LSTM) · Gated recurrent unit (GRU) · And convolutional neural networks (CNN) · Word2Vec · GloVe · fastText

1 Introduction

Sarcasm is a characteristic sentiment where feelings are expressed using intensified positive or positive words, typically intended to bring forth a negative connotation of the written text. It is a complex semantic tool commonly utilized in online client produced content, and as a type of humor evoking mechanism while communicating a supposition. Consequently, to interpret the sarcastic content correctly it is essential to ignore the contextual sense of the text and rather understand the suggested (naturally inverse) sense. Sarcasm represents a critical challenge in the domain of Natural Language Processing (NLP) [8, 42], especially in the sentimental analysis. The challenge is the work of deriving sentiment polarity of content - whether the author is agreeable to, or against, a particular subject.

At the point when sarcasm is available in the content, it can alter the opinion extremity of positive or negative, as sure sounding expressions can have particular negative importance. Various organizations regularly use sentimental analysis to measure general conclusions on their items and services. In any case, exemplary sentiment analyzers are unable to detect a verifiable significance of the sarcastic content and incorrectly characterize the judgement provided by the author. Developments in programmed sarcasm recognition, research can improve the engagement of sentiment analysis [5, 64] vastly. Therefore, any program that endeavors in deciding the importance of the client produced text precisely should be equipped for identifying sarcasm.

Sarcasm is a type of phenomenon with specific perlocutionary effects on the hearer, such as to break their pattern of expectation. Consequently, correct understanding of sarcasm often requires a deep understanding of multiple sources of information, including the utterance, the conversational context, and, frequently some real-world facts. The most significant step in sarcasm detection tasks is to precisely decide the accuracy of the statements from an exacting perspective, to arrange text dependent on an extremity of expressed emotion (positive or negative). This yields good outcomes on account of factual language since it passes on the standard interpretation. Nonetheless, the utilization of metaphorical language which is naturally significant speaks to some different options from the undeniable importance, accordingly making sentiment investigation a non-trivial issue. However, the utilization of figurative language [32] constitutes something different from the existing meaning which makes sentimental analysis a non-trivial issue.

Sarcasm [16, 43] shows the informed disunity between the real circumstance and the articulation of the same content. For example, a text that says, “The wonderful feeling of spending hours stuck in a traffic jam!” unmistakably demonstrates this friction between the genuine circumstance of “being stuck in a traffic jam” and the articulation content “wonderful.” This differentiation and move of emotions in sarcastic instances depict sarcasm as a

specific occasion of sentimental analysis. Subsequently, the programmed sentimental analysis of vast and different online content will be improved by discovery of similar comments and texts.

Primitive models developed for sarcasm detection relied on the very low and primitive features for the detection such as the number of tokens in a sentence. In our approach, we are proposing an ensemble model that combines the features of deep learning algorithms [7] like CNN, LSTM and GRU. Our approach also uses various word embeddings for vectorization of each token in the dataset. The results that we got, clearly depict that word embeddings significantly improved the accuracy of our proposed model. The ensemble model used has tried to bring down the gap between learning of unique and temporal highlights of the textual information.

1.1 The Objectives of the paper are:

- In this research, we proposed an ensemble model using Baseline CNN, Bi-Directional LSTM and GRU which will recognize, learn sarcastic patterns and will provide improved accuracy. This ensemble model helped us to build up a viable sarcasm identification solution.
- Our proposed model has been pre-trained word embedding models like Word2Vec, GloVe, fastText and compared their accuracies which enhanced the precision of the proposed model.
- After training and validating our proposed model on two publicly available datasets, we have found that our approach not only improved the detection of sarcasm but also provides overall improvised insights.
- Our proposed model is flexible and accurate since we are using an ensemble of neural network models, this model once trained on one dataset can easily work on the other sarcasm datasets to provide the precise, and accurate results.
- The results provide solid insights (in terms of the accuracy of the proposed model) for the system developers to integrate the proposed model into real-time analysis of any review or comment posted in the public domain.

The remaining part of the paper is split into 6 sections- Section 2 explains the background and related research done in this domain. Section 3 explains the datasets used on which the model has been tested. Section 4 discusses the methodology including the deep learning methods that are being used to detect sarcasm. Section 6 has been dedicated to the results and analysis after training the model using the proposed methodology. In the end, section 7 concludes this research describing the future scope related to this study.

2 Background and related work

The increasing pursuit of Internet users in all types of social media has strengthened researchers' interest to keenly mine the content accessible, both quantitatively and subjectively. The catchphrase "sentiment analysis" was at first witnessed in the published work [13] in 2003, and from that point forward, both primary [10, 28, 60] and secondary researches have been presented across pertinent literature [1, 29–31]. Besides, the literature is well-equipped with research related to sentiment and sarcasm analysis using machine learning and deep

learning paradigms on specifically textual user-generated online content on social media. Aloufi et al. [2] proposed a model for sentiment analysis of football explicit tweets using three classifiers, specifically, support vector machines, multinomial Naive Bayes and random forests. Pai et al. [51] presented a model for prediction of vehicle deals by sentiment analysis of twitter tweets and stock market valuation using least square support vector regression. Research using deep learning models for sentiment analysis has also been reported. Tseng et al. [63] detected textual opinions found in teaching evaluation questionnaires and applied the analysis results in order to assist the choice of remarkable teaching faculty members using attention-based LSTM.

Wu et al. [65] proposed a model having quadratic associations of LSTM capable of catching complex semantic representations of common language texts and assessed on the benchmark dataset, the Stanford Sentiment Treebank. Bouazizi et al. [10] broadened the concept of binary or ternary classification and presented an approach to classify texts collected from Twitter site into seven sentiment-based classes. The researchers further proposed [11] multi-class sentiment analysis which tends to address the identification of the exact sentiments shown by the users utilizing the tasks of evaluation that recognizes all the existing sentiments inside a tweet instead of providing a solitary sentiment labelling to it. Felbo et al. [14] put forth a DeepMoji model, which depends on the occurrences of emoticons, for the task of detecting the emotional information on Twitter. It utilizes a variation of LSTM, a 6-layer model that is a hybrid model of BiLSTM and the attention component for the identification of sarcastic tweets.

Ghosh et al. [17] in 2016, suggested a neural organization semantic technique for the assignment of sarcasm identification. They have additionally proposed semantic models utilizing Support Vector Machines (SVM) which uses constituency parse-trees marked with semantic and syntactic data. The proposed model surpasses best in class text-based strategies for sarcasm identification, yielding 0.92 as F-score. Amir et al. [4] in 2016 proposed a model that would adapt consequently and afterwards gained by user embeddings, utilizing working together with contextual attributes naturally from history tweets. Experimental results showed that when compared with discrete manual features, neural characteristics give better precision for sarcasm identification, with various mistake conveyances. Their model gave better outcomes over the best in the class discrete model.

Hazarika et al. [19] in 2016 developed models which depended on a previously trained convolutional neural network framework for extricating personality, emotions, and sentimental features for sarcasm identification. Such characteristics permitted the proposed models to beat the best in class on standard datasets alongside the network's baseline features. Ghosh et al. [18] in 2017 concentrated on social networking platform discussions, the authors investigated these issues: Does modelling of discussion context help in sarcasm identification and would we be able to comprehend what part of discussion context set off the sarcastic reverts. To address the primary issue, they examined different sorts of Long Short Term Memory (LSTM) networks which could show the sarcastic reaction and the discussion context. Mishra et al. [48] in 2017 introduced a model to naturally obtain insight characteristics using the eye-motion / gaze information of human behavior by studying the content and utilizing them as aspects alongside literary characteristics for the tasks of sentimental polarity and sarcasm identification. Attributes from both text and gaze were taken and utilized by CNN to characterize the captured content. Porwal et al. [56] in 2018 aimed at using a recurrent neural network (RNN) approach for sarcasm detection because it automatically extracts features required for machine learning approaches. Along with the RNN, their methodology also uses Long Short-Term

Memory (LSTM) technique on TensorFlow to identify semantic and syntactic data over Twitter tweets dataset to identify sarcasm.

Mehndiratta et al. [44] in 2019 attempted to assess different AI models alongside standard and hybrid deep learning techniques across other normalized datasets. Authors utilized word embedding techniques to perform vectorization of text. They employed three normalized datasets accessible in the open-source realm and utilized various word embeddings, i.e., Word2Vec, GloVe, and fastText, to approve the theory. The primary finding was the hybrid model which incorporates Convolutional Neural Network (CNN), and Bidirectional Long Term Short Memory (Bi-LSTM) beats other ordinary machine learning and deep learning techniques over various datasets observed in the research. This made the proposed theory valid. Pelser et al. [53] in 2019 proposed a profound 56-layer organization, actualized with dense connectivity to display the detached articulation and concentrate more on extravagant characteristics in that. They differentiated their methodology against ongoing best in class models which utilizes irrelevant content, and exhibit competitive outcomes while using just the local features of the content. a contextual analysis was also introduced, supporting their methodology precisely characterizing different uses of apparent sarcasm, which a benchmark CNN misclassified.

Jain et al. [21] in 2020 presented a model which is used to detect sarcasm in bi-lingual comprising of English and Hindi concoction tweets is a blend of bidirectional long short-term memory with soft attention technique and characteristic-rich convolution neural network prepared utilizing an amalgamation of Hindi, English, and additional pragmatic characteristic vectors. This model includes three modules which are English, Hindi processing module, and the classifier module, which is employed to get the output predictions. Kumar et al. [34] in 2020 introduced a Multi-head Attention based bidirectional long short-term memory (MHA-BiLSTM) model used to recognize sarcastic text in the stated dataset. Experimental outcomes revealed that a multi-head attention model improved precision of BiLSTM, and it performed superior than characteristics-rich SVM techniques.

The previous researchers utilized various machine learning and deep learning models along with pre-trained word embeddings to improve the precision of their models. The past studies mainly utilized twitter's tweet datasets for detection of sarcasm on social media platforms. The dataset's size previously used was not enough to provide conclusive sarcasm detection methodology. Hence, by observing the past work and in order to enhance our model, we have used the dataset from another popular social media platform namely Reddit and online news platform's dataset. In the past, there were not any studies using the ensemble models for sarcasm detection, by using ensemble models we were able to overcome the limitations such as overfitting and underfitting of the individually trained models. Ensemble models used in our study not only improvised stability but also accuracy and predictive power of the proposed model.

3 Materials and methods

3.1 Materials – Dataset description

3.1.1 News headlines

The news headlines [49] dataset used to detect sarcasm, is taken from two prime news websites specifically the Onion and the HuffPost. Events of the sarcastic type were obtained from The

Onion and the non-sarcastic events were obtained from The HuffPost. Here, Fig. 1 shows the dataset's snippet, Figs. 2 and 3 shows the word cloud for sarcastic and non-sarcastic comments respectively. The advantages of including these datasets enables reduction in sparsity and an increase in the possibility of obtaining pre-trained embeddings, collection of good quality labels with lesser noise in contrast to twitter datasets [6]. Additionally, this data set is self-contained. The content of this dataset comprises of three attributes:

- is_sarcastic- if the document is sarcastic then this value is 1, else it is 0.
- headline- it provides the news article headlines.
- article_link- it gives the original news article links which is helpful in gathering additional information.

3.1.2 Sarcasm on Reddit

This dataset [25] was produced by scraping a large set of comments from Reddit which comprises sarcastic comments from Internet commentary. The data consists of balanced and imbalanced versions in the ratio 1:100. The corpus includes 1 million sarcastic statements and the response of many non-sarcastic comments from the same source. Here, Figs. 4 and 5 shows the dataset's snippet for sarcastic and non-sarcastic comments respectively., and Fig. 6 shows the word cloud for sarcastic comments in the dataset. Table 1 provides the description about the dataset size used for training and testing the models.

	article_link	headline	is_sarcastic
0	https://www.huffingtonpost.com/entry/versace-b...	former versace store clerk sues over secret 'b...	0
1	https://www.huffingtonpost.com/entry/roseanne-...	the 'roseanne' revival catches up to our thorn...	0
2	https://local.theonion.com/mom-starting-to-fea...	mom starting to fear son's web series closest ...	1
3	https://politics.theonion.com/boehner-just-wan...	boehner just wants wife to listen, not come up...	1
4	https://www.huffingtonpost.com/entry/jk-rowlin...	j.k. rowling wishes snape happy birthday in th...	0

Fig. 1 News headline dataset's snippet

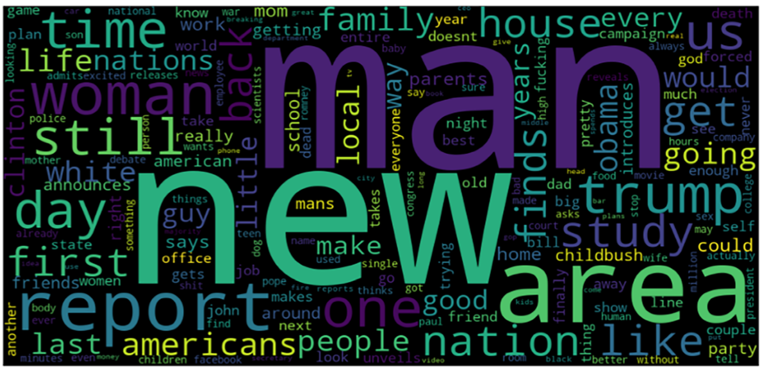

Fig. 2 Word Cloud for sarcastic comments in news headlines dataset

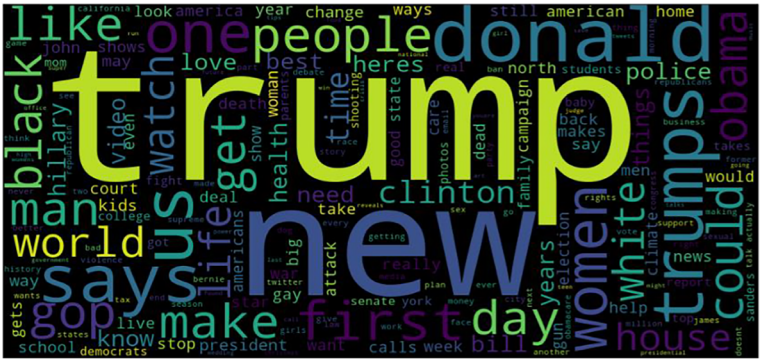

Fig. 3 Word Cloud for non-sarcastic comments in news headlines dataset

	label	comment	author	subreddit	score	ups	downs	date	created_utc	parent_comment
0	1	but they will have all those reviews	RoguishPoppet	ProductTesting	0	-1	-1	2016-11	2016-11-01 02:04:59	The dumb thing is, they are risking their sell...
1	1	wow it is totally unreasonable to assume that ...	pb2crazy	politics	2	-1	-1	2016-11	2016-11-01 02:42:11	Clinton campaign accuses FBI of 'blatant doubl...
2	1	ho ho ho but melania said that there is no way...	pb2crazy	politics	8	-1	-1	2016-10	2016-10-18 16:20:53	Anyone else think that it was interesting the ...
3	1	i can not wait until potus starts a twitter wa...	kitduncan	politics	3	-1	-1	2016-11	2016-11-01 03:22:33	Here's what happens when Obama gives up his Tw...
4	1	gotta love the teachers who give exams on the ...	DEP61	CFBOffTopic	3	-1	-1	2016-11	2016-11-01 03:30:11	Monday night Drinking thread Brought to You by...

Fig. 4 Snippet of sarcastic comments in Sarcasm on Reddit Dataset

3.2 Methodology

The proposed model aims to detect sarcasm using deep learning techniques [44, 46]. This segment describes the numerous techniques applied on the datasets described in section 3. The experimentation in this study uses an ensemble model [38] along with word embeddings in order to detect sarcasm. The ensemble model once compared which include LSTM, CNN, GRU and applied for the overall precision of the methods are used on the datasets considered in section 3. Prior to the real classification of the data, various steps have been performed for data pre-processing and finally for training data. The study utilized the conventional 80:20 split for training and validation purposes respectively. The flow chart given in Fig. 7 describes the major steps used in our proposed methodology for sarcasm detection.

3.2.1 Data preprocessing

Data Preprocessing [21, 27, 62] modifies the raw dataset into a comprehensible format. It enhances the data efficiency which affects the result of algorithms. A number of filters have

	label	comment	author	subreddit	score	ups	downs	date	created_utc	parent_comment
0	0	NC and NH.	Trumpbart	politics	2	-1	-1	2016-10	2016-10-16 23:55:23	Yeah, I get that argument. At this point, I'd ...
1	0	You do know west teams play against west teams...	Shbshb906	nba	-4	-1	-1	2016-11	2016-11-01 00:24:10	The blazers and Mavericks (The wests 5 and 6 s...
2	0	They were underdogs earlier today, but since G...	Creepeth	nfl	3	3	0	2016-09	2016-09-22 21:45:37	They're favored to win.
3	0	This meme isn't funny none of the "new york ni...	icebrotha	BlackPeopleTwitter	-8	-1	-1	2016-10	2016-10-18 21:03:47	deadass don't kill my buzz
4	0	I could use one of those tools.	cush2push	MaddenUltimateTeam	6	-1	-1	2016-12	2016-12-30 17:00:13	Yep can confirm I saw the tool they use for th...

Fig. 5 Snippet of sarcastic comments in Sarcasm on Reddit Dataset

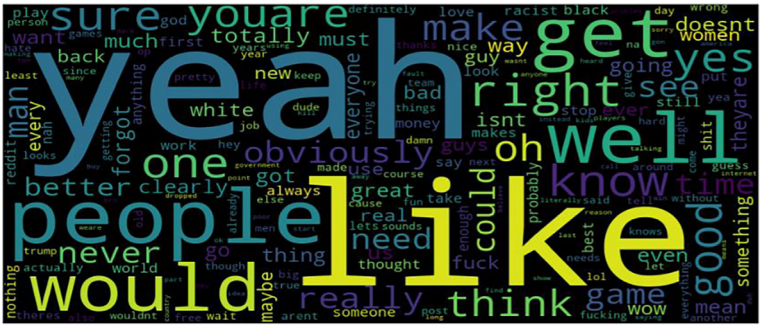

Fig. 6 Word Cloud for sarcastic comments in Sarcasm on Reddit Dataset

Table 1 Dataset description

Dataset	Training		Testing	
News headlines	20,287	23,976	5071	5994
Sarcasm on reddit	449,962	449,993	49,995	49,999

been applied on the news headlines dataset and comments scraped from Reddit in order to get an understandable format. Natural language toolkit (NLTK) has been used prior to the training stage for preprocessing both the datasets. To preprocess the datasets various steps were taken including concatenation of content, elimination of duplicate text, employment of tokenization to get individual tokens, replacing non-essential content like white spaces, URL addresses, brackets, stopwords, colons. Finally, lemmatization to extract lemma of a given word after its morphological analysis was completed and the dataset ready for further study.

3.2.2 Word embeddings

Word embeddings [22, 45, 50] is a method of representing words in the numeric vector form. It learns a divided representation for a vocabulary from a collection of text. It possesses the ability to divulge several unknown relationships between the words. It is an advancement over the traditional ways to represent words, for instance, the bag-of-words model which generates immense sparse vectors which are computationally not possible to represent the complete vocabulary. In this, the categorization of similar words together takes place. Three desired methods of learning word embeddings from the data incorporate Word2vec, GloVe and fastText [52, 57]. Word embeddings add contextual meaning to the words that have been selected and would be vectorized. The vectorization of the word is accomplished by using the above embedding techniques. The above pretrained models allow association of a word to a specific meaning and thereby encoding in binary form. The binary digit 1 encodes presence of sarcasm in text while the digit 0 is used when sarcasm is not detected. The data collected is then analyzed and further output is derived.

The idea of the word embeddings is that if there is a user in the training data with a certain personality, and they happen to make sarcastic tweets, then when we get new data and there is a new user that has a similar style and therefore similar embedding to the previous user,

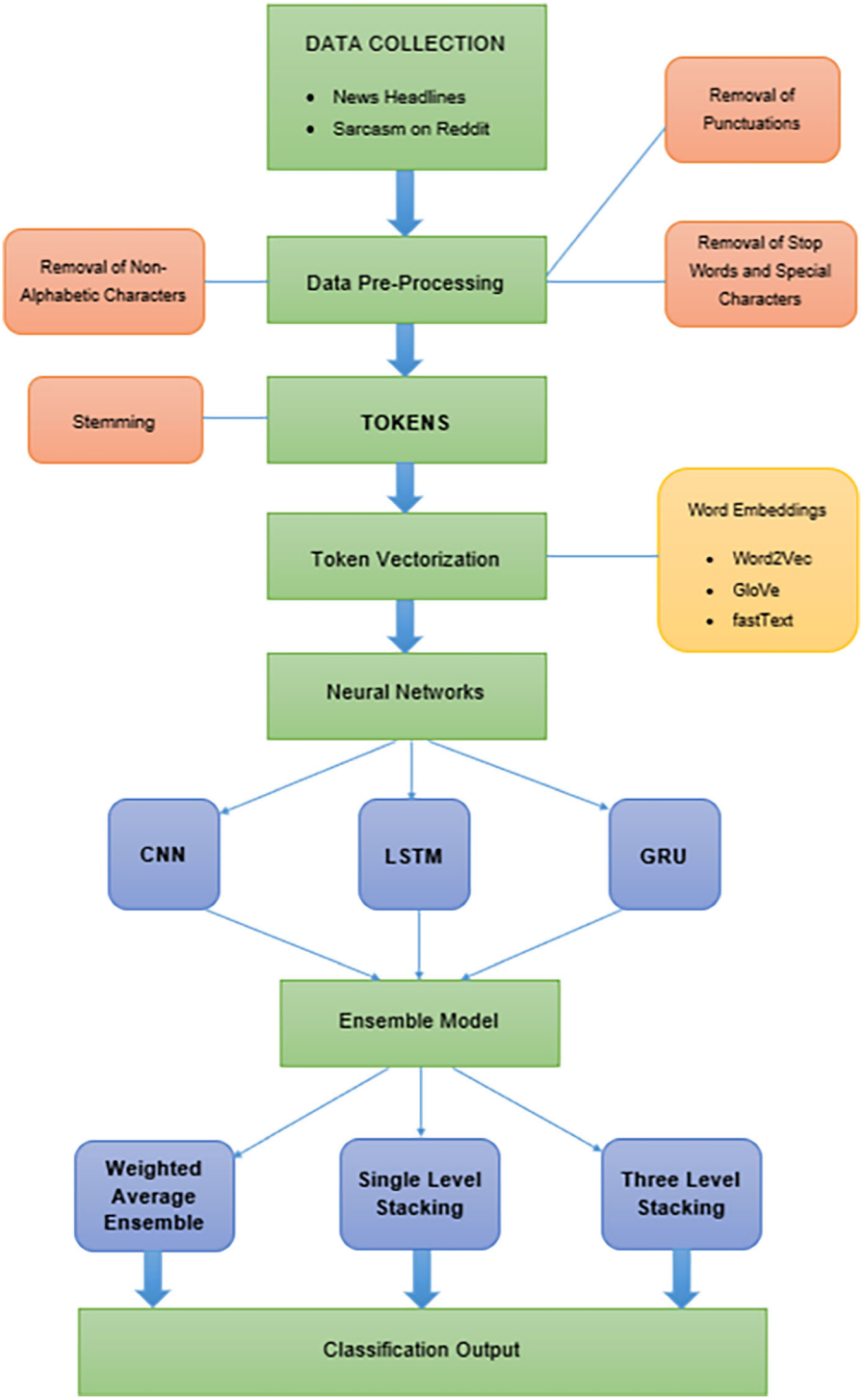

Fig. 7 Methodology flow chart

without looking at the new user's tweet, we can predict if this user will be sarcastic or not, just by looking at the similarity of the embeddings.

Word embeddings allow a more productive and quicker output, permitting to train and learn better from a huge corpus. This technique enables sharing the representation across words which helps in creating more stable representation of words which is quite rare. We utilized the openly accessible Word2Vec [23, 39, 47] vectors, which are pre-trained on 100 million words available on Google News [55] having 300 dimensional vectors, GloVe [54] embeddings, which are trained on Common Crawl and Wikipedia datasets, and fastText [9, 24, 26] word-embeddings having 1 million-word vectors trained on Wikipedia 2017 having 300 dimensional vectors.

3.2.3 Deep learning frameworks

Deep learning [27, 44, 66] is a class of machine learning which shows better results on unstructured data. Deep learning permits computational models to continuously learn from the given data and implement classification tasks from the provided text, sound or images. Deep learning models can attain state-of-the art precision which occasionally even exceed the human level of execution. Models are trained by making use of neural network architectures which consist of many layers and huge sets of labeled data.

For Sarcasm detection, we have incorporated deep learning techniques such as CNN, LSTM, and GRU. Here, Fig. 8 depicts the LSTM architecture for sarcasm detection. We have initially passed the pre-processed input sequence to the pretrained word embedding layer which helps in formation of fixed length vectors by assigning a unique index to each and every word in a given sentence. Following this, a layer of Bidirectional LSTM is applied to extract long distance reliance across the content, pooling layer integrates the collected information to pool a feature dimension which is transformed to a column vector through a flatten layer. Softmax executes a classification which completes the whole neural network process. Similarly, we have trained CNN & GRU models to detect sarcasm from our input. A detailed algorithm is provided below in order to depict how our models are detecting sarcasm through individual deep learning models. These individually trained models are then fed to ensemble model as input so as to achieve more precise ensemble model.

Convolutional neural network (CNN) Convolutional neural network (CNN) was initially developed by LeCun in [37] for classification of handwritten numbers. When we applied CNN

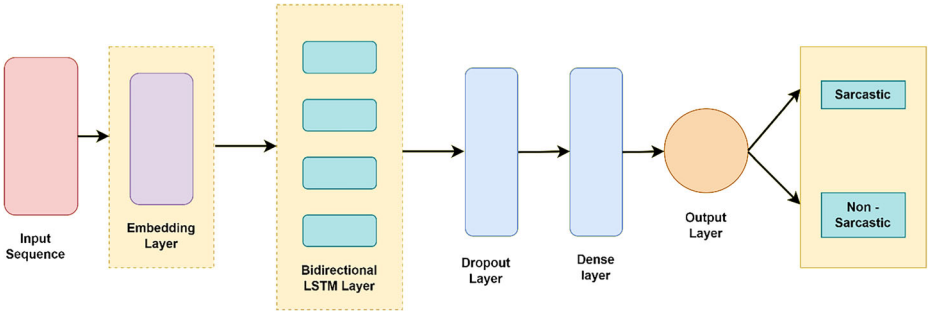

Fig. 8 Sarcasm detection LSTM architecture

on our content the model learnt, extracted features and detected patterns. The implementation of CNN is made up of five vital layers namely The Convolutional Layer, Pooling or Down-sampling Layer, Dense Layer, Flattening Layer, Fully Connected Layer.

Long short-term memory neural network (LSTM) Long Short-Term Memory Neural Network (LSTM) was first presented by Hochreiter and Schmidhuber in [20] later it was enhanced and implemented to a vast diversity of problems.

In this, 300 LSTM cells were applied with a single layer of LSTM. Four individual and independent calculations were carried out with the help of four gates for each cell.

Gated recurrent unit (GRU) Gated Recurrent unit (GRU) was presented by Cho in [12] which has the similar architecture as LSTM. Gated Recurrent unit (GRU) [41] used here aims to help connection through a series of nodes to execute machine learning tasks related with memory and clustering, for example in text identification. GRU is used here to modify neural network input weights to solve the vanishing gradient problem that is a frequent issue with recurrent neural networks.

Algorithm 1: Training the models

Input: News Headline and Sarcasm on Reddit Datasets

Output: Sarcasm label

Result: All trained models are obtained

```
while Untested model is available do
    initialize model: import the architecture, add the required layers;
    if Model is not trained then
        load dataset;
        label text data with the meta data available;
        append text data path to the dataframe;
        convert the text data at path into numpy array;
        train (80%), validation (10%), and test (10%) dataset;
        train model with the assigned architectures (CNN, LSTM, GRU);
        save trained model;
    else
        load trained model;
    end
    use test dataset to test the trained model;
    use validation set to evaluate model;
end
```

3.2.4 Proposed model - ensemble learning

Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would. Ensemble Learning [38, 40] is associated with learning how to best consolidate forecasts from different pre - existing models (known as base-learners). Every member of the ensemble makes a contribution to the final resultant and discrete weaknesses are neutralized with the help of contributions done by other members. The meta-learner is the combined learned model. We have implemented Ensemble learning [15] to enrich the performance of our models with regard to prediction, classification, function approximation etc. We have tried to create a bootstrap aggregated model using the following three pre-trained ensemble models using the DeepStack library. Deep Stack is a python package used to build deep learning ensemble models, originally built on top of Keras and distributed under the MIT license. Here, we are leveraging the power of both neural networks and the ensemble models to enhance the precision of our proposed model. The proposed model firstly gets trained on each neural network model individually, and then these individually trained models are embedded into the Ensemble Models to get the best out of these techniques.

- **Single Level Stacking:** Stacking is based on training a Meta-Learner on top of pre-trained Base-Learners. DeepStack offers an interface to fit the Meta-Learner on the predictions of the Base-Learners. In this, we have used scikit-learn library to create a single level stacking model, the output which we received from base learners was used to provide input to the meta-learner which learns in a way to combine the base learners' predictions resulting in enhanced output. The architecture of the single level stacking ensemble model based on top of pre-trained Keras models is shown in Fig. 9.
- **Three-Level Stacking:** There is no limitation to the number of stacked levels in stacked generalization. With the help of scikit-learn Stacking interface a 3rd level meta-learner with DeepStack was utilized in order to achieve a more powerful model than single level stacking. The architecture of the three-level stacking ensemble model based on top of pre-trained Keras models is shown in Fig. 10.

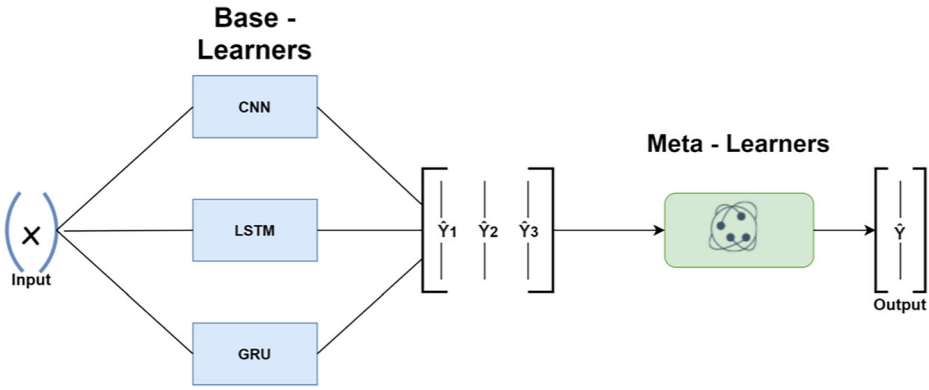

Fig. 9 Single level stacking ensemble model

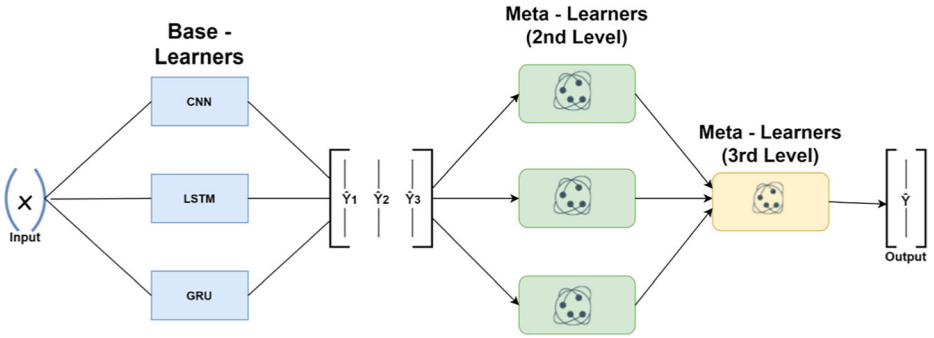

Fig. 10 Three-level stacking ensemble model

Here, the basic idea is to train deep learning algorithms individually with training dataset and then generate a new dataset with these models. Then this new dataset is used as input for the combined deep learning algorithm and this process can continue for further levels as per need. We can call this model as level-based stacking ensemble model.

- **Weighted Average Ensemble:** In this, Weighted Average Ensemble Technique weights the prediction of each ensemble member, combining the weights to calculate a combined prediction. Weight optimization search is performed with randomized search based on the Dirichlet distribution on a validation dataset. We have added our previously trained models to the Dirichlet Ensemble object, then the model is fitted using the Dirichlet Markov Ensemble method and its resultant accuracy was obtained. We optimized the weights which were utilized for weighing every output received from base-learners along with considering weighted average. There was no meta-learner used in this ensemble. The architecture of the weighted average ensemble model used in our proposed model is shown in Fig. 11.

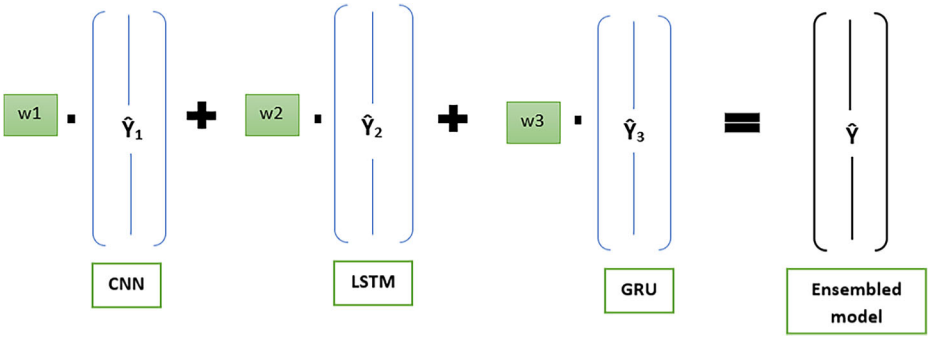

Fig. 11 Weighted average ensemble model

A detailed algorithm is provided below to depict how Weighted Average ensemble model approach is being used to detect sarcasm.

Algorithm 2: Creating Ensemble Model

Input: Trained neural networks

Output: Ensemble neural network Model

```
import deepstack library;
create DirichletEnsemble object;
while non - bootstrapped model is available do
    if Model is not bootstrapped then
        load model;
        create train, test, validation data set (80%,10%,10% respectively);
        create an object of class kerasMember passing the model and datasets as argument;
        add the kerasMember object to DirichletEnsemble object;
    else
        Fit DirichletEnsemble object;
        calculate weights for all the models;
    end
    compute Accuracy from prediction scores;
    print the final weights of each model;
    print the final accuracy obtained;
end
```

4 Result analysis

In this segment, we will discuss the different techniques that we have applied to the dataset and the result acquired by it. The purpose of the proposed methodology is to detect sarcasm [58, 59] in textual data using deep learning techniques accurately. We have not only used deep learning algorithms like CNN, LSTM and GRU [3, 21, 33, 35], but an ensemble model has also been proposed that combines the features of the above-mentioned methods. The ensemble models [61] which we have used for our research are Weighted Average Ensemble, Single Level Stacking and Three-Level Stacking and these have been compared and applied for accuracy of techniques on both the datasets used. From the results, we were able to draw an inference that the Weighted Average Ensemble gives the highest accuracy for both the datasets used. The ensemble model used has tried to bring down the gap between learning of unique and temporal highlights of the textual information [36]. The model has been trained using Sigmoid as the activation function and Adam as the optimizer with dropouts of 0.15, 0.25 and 0.35 respectively. The performance of the model was enhanced using parameters and hyperparameters [32] respectively, with the parameter settings listed in Table 2.

Table 2 Parameters list for Training and Validating our models

Parameters	Values
Filters	64
Kernel	3
Embedding Dimension	300
Epochs	30
Activation Function	Sigmoid
Loss	Binary Cross entropy
Batch Size	64/128
Word Embedding	Word2Vec, GloVe, FastText
Pool Size	2
Dropouts	0.15, 0.25, 0.35
Optimizer	Adam

4.1 News headlines dataset

The described model was trained on 80% of the data in the news headlines dataset (20287-Sarcastic comments and 23,976-Non-Sarcastic comments) and tested on 20% of the data (5071-Sarcastic comments and 5994-Non-Sarcastic comments). The model has been executed for 30 epochs having a batch size of 64 and the training curves for the CNN, LSTM and GRU models with GloVe word embedding have been depicted in Figs. 12, 13 and 14 respectively.

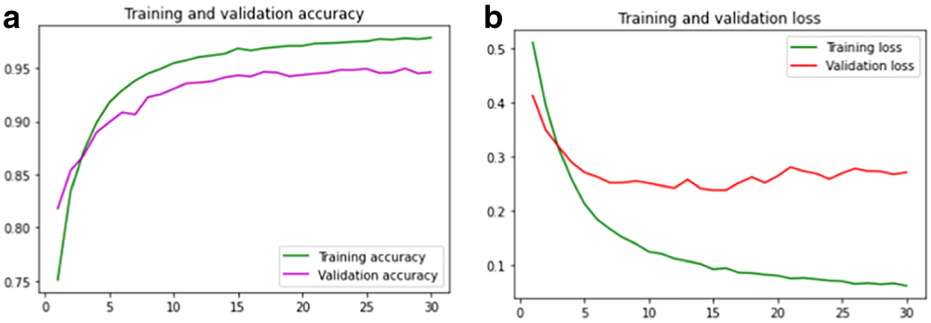

Fig. 12 **A:** Training Accuracy (in green) and Validation Accuracy (in pink) vs Number of Epochs of CNN Model with GloVe word embeddings. **B:** Training Loss (in green) and Validation Loss (in red) vs Number of Epochs of CNN Model with GloVe word embeddings

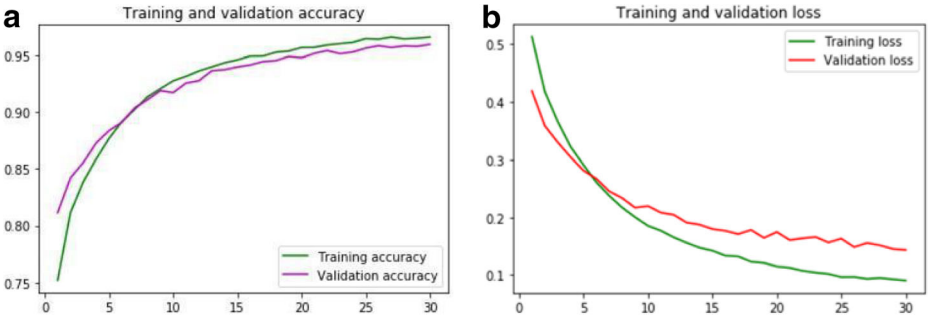

Fig. 13 **A:** Training Accuracy (in green) and Validation Accuracy (in pink) vs Number of Epochs of LSTM Model with GloVe word embeddings. **B:** Training Loss (in green) and Validation Loss (in red) vs Number of Epochs of LSTM Model with GloVe word embeddings

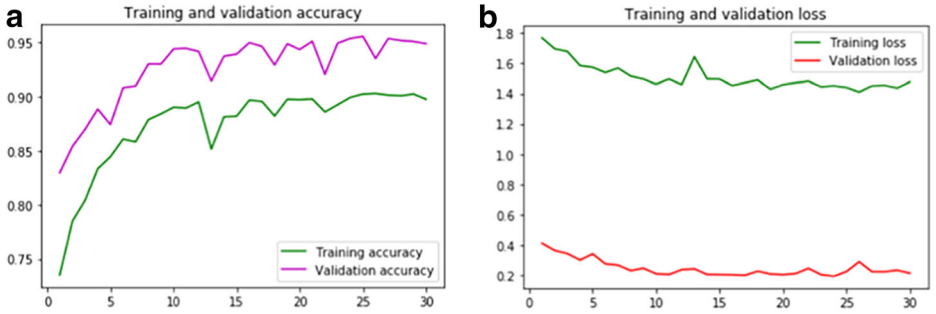

Fig. 14 **A:** Training Accuracy (in green) and Validation Accuracy (in pink) vs Number of Epochs of GRU Model with GloVe word embeddings. **B:** Training Loss (in green) and Validation Loss (in red) vs Number of Epochs of GRU Model with GloVe word embeddings

Table 3 exhibits the accuracy obtained on testing the model on the news headlines dataset while Fig. 15 depicts the graphical representation for the same. It can be observed that without word embeddings, accuracy obtained for CNN, LSTM and GRU model is 94.28%, 95.12% and 94.47% respectively while with word embeddings, GloVe performed the best with accuracy obtained as 95.36%, 96.10% and 95.64% respectively for CNN, LSTM and GRU model. Table 4 depicts the accuracy percentage of the ensemble model while Fig. 16 depicts

Table 3 News headlines dataset accuracy

News headlines - accuracy (%)			
Type of word embeddings	CNN	LSTM	GRU
None*	94.28	95.12	94.47
Word2Vec	94.37	94.83	94.63
GloVe	95.36	96.10	95.64
FastText	94.6	95.72	95.4

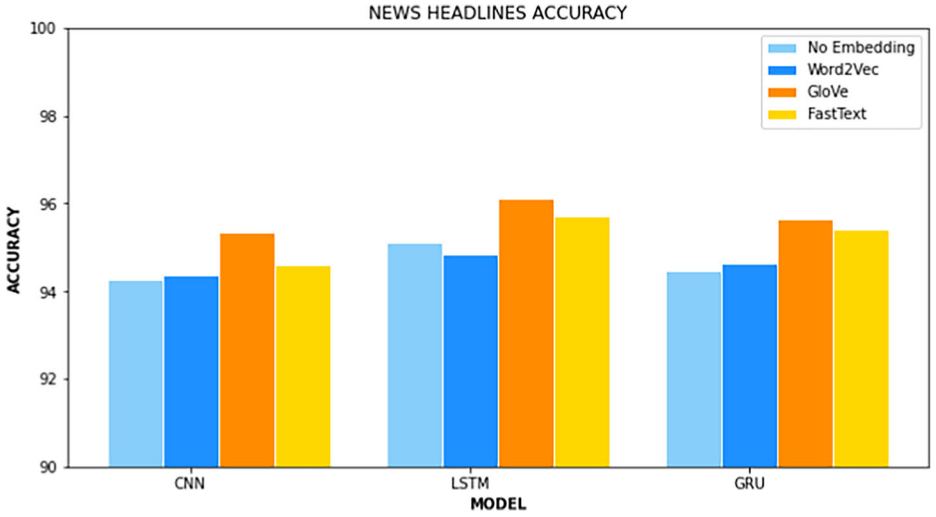

Fig. 15 News headlines accuracy

Table 4 News headlines dataset ensemble models accuracy

News headlines – ensemble model accuracy (%)			
Type of word embeddings	Weighted average	Single level stacking	Three-level stacking
None*	98.09	94.68	94.7
Word2Vec	98.26	94.85	95.22
GloVe	98.97	96.21	96.4
FastText	98.6	95.9	95.94

*Here, None represents model trained without any pre-trained word embeddings

the graphical representation for the same. It can be concluded from the Table 4 that the ensemble model with Glove word embedding has performed the best when compared to other word embeddings and has achieved a weighted average of 98.97% while the single-level stacking and three-level stacking has the values as 96.21% and 96.4% respectively. It can also be noticed that the LSTM technique has obtained an elevated accuracy as compared to the CNN and GRU models with or without word embeddings.

4.2 Sarcasm on reddit dataset

The described model was trained on 90% of the images in the Reddit dataset (449962-Sarcastic comments and 449,993- Non-Sarcastic comments) and tested on 10% of the images (49995-Sarcastic comments and 49,999- Non-Sarcastic comments). The model has been executed for 10 epochs having a batch size of 128 and the training curves for the CNN, LSTM and GRU models with GloVe word embedding have been depicted in Figs. 17, 18 and 19 respectively. Table 5 shows the accuracy obtained on testing the model on the Reddit dataset while Fig. 20 depicts the graphical representation for the same. It can be observed that without word embeddings, accuracy obtained for CNN, LSTM and GRU model is 71.48%, 71.82% and 71.67% respectively while with word embeddings, Word2Vec performed the best

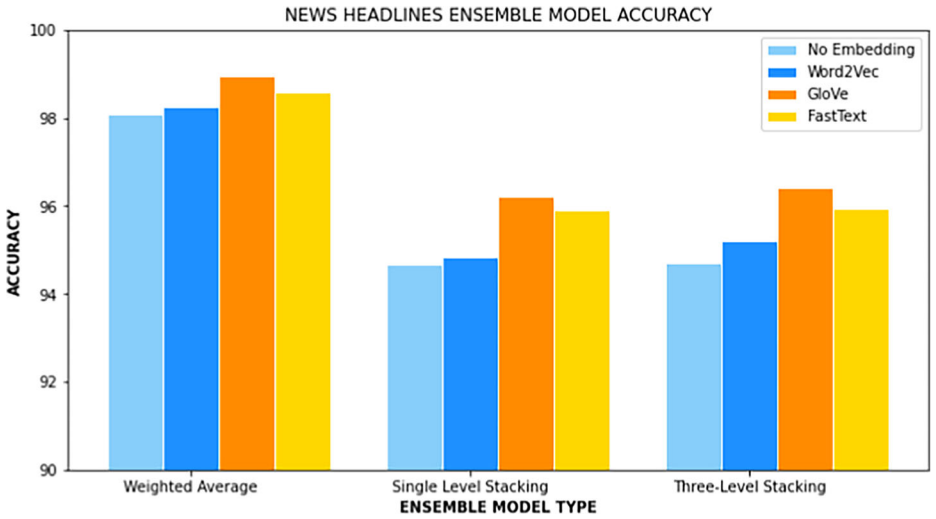

Fig. 16 News headlines ensemble model accuracy

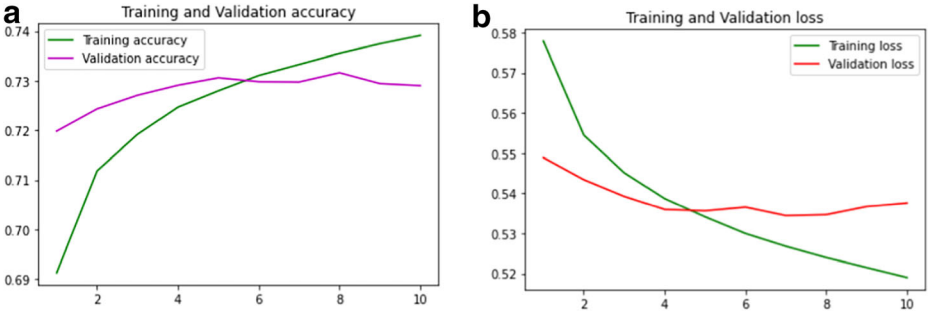

Fig. 17 **A:** Training Accuracy (in green) and Validation Accuracy (in pink) vs Number of Epochs of CNN Model with Word2Vec word embeddings. **B:** Training Loss (in green) and Validation Loss (in Red) vs Number of Epochs of CNN Model with Word2Vec word embeddings

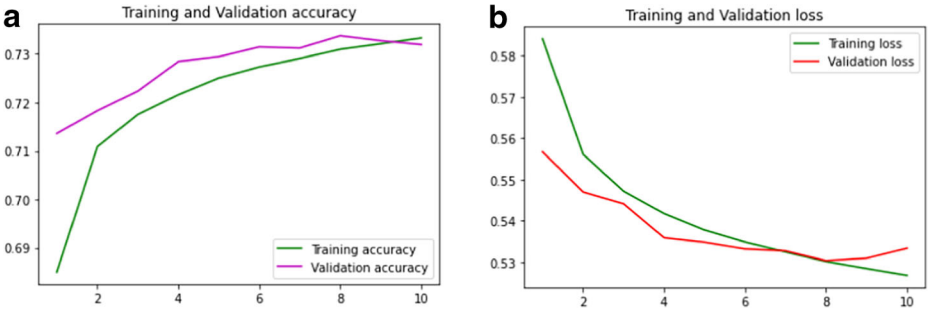

Fig. 18 **A:** Training Accuracy (in green) and Validation Accuracy (in pink) vs Number of Epochs of LSTM Model with Word2Vec word embeddings. **B:** Training Loss (in green) and Validation Loss (in Red) vs Number of Epochs of LSTM Model with Word2Vec word embeddings

with accuracy obtained as 73.19%, 73.65% and 73.34% respectively. Table 6 depicts the accuracy percentage of the ensemble model while Fig. 21 depicts the graphical representation for the same. It can be concluded from the Table 6 that the ensemble model with Word2Vec word embedding has performed the best when compared to other word embeddings and has achieved a weighted average of 81.64% while the single-level stacking and three-level

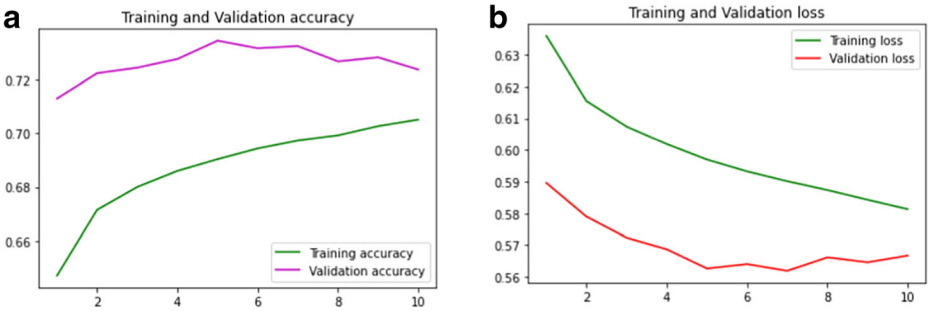

Fig. 19 **A:** Training Accuracy (in green) and Validation Accuracy (in pink) vs Number of Epochs of GRU Model with Word2Vec word embeddings. **B:** Training Loss (in green) and Validation Loss (in Red) vs Number of Epochs of GRU Model with Word2Vec word embeddings

Table 5 Sarcasm on reddit dataset accuracy

Sarcasm on reddit - accuracy (%)			
Type of word embeddings	CNN	LSTM	GRU
None*	71.48	71.82	71.67
Word2Vec	73.19	73.65	73.34
GloVe	72.02	72.48	72.26
FastText	73.15	73.43	73.21

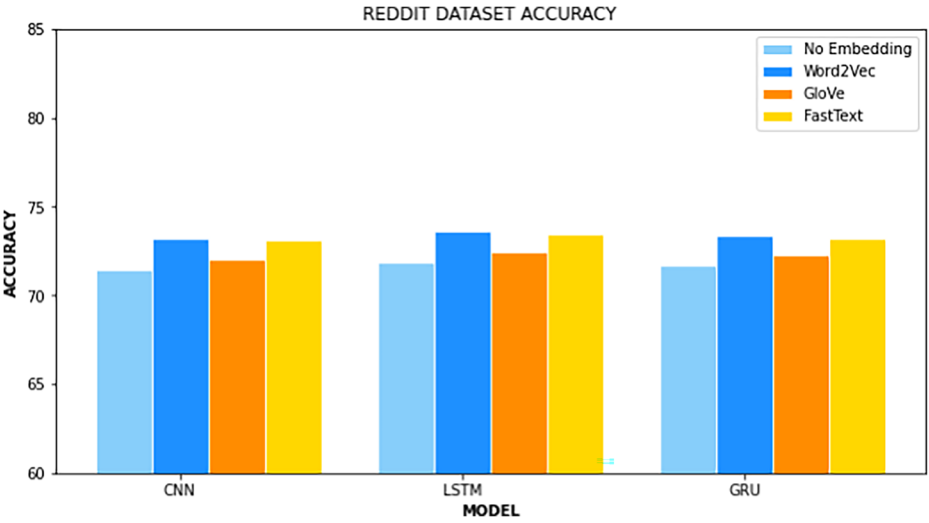

Fig. 20 Reddit dataset accuracy

stacking has the values as 73.85% and 73.92% and respectively. It is evident from the results that the LSTM technique has obtained an elevated accuracy percentage than the CNN and GRU models with or without word embeddings.

We observed that word embeddings perform a major part when we need to execute a task linked with natural language processing utilizing deep learning. The word embeddings that we made use of in our research includes Word2Vec, fastText, and GloVe. Some of the conclusions that we concluded from the above tables are vital for us to perceive the conduct of our proposed framework and its performance too. The above study depicts that the proposed

Table 6 Sarcasm on reddit dataset ensemble models accuracy

Sarcasm on reddit – ensemble model accuracy (%)			
Type of word embeddings	Weighted average	Single level stacking	Three-level stacking
None*	80.27	69.83	70.02
Word2Vec	81.64	73.85	73.92
GloVe	80.75	73	73.21
FastText	81.13	73.56	73.69

*Here, None represents model trained without any pre-trained word embeddings

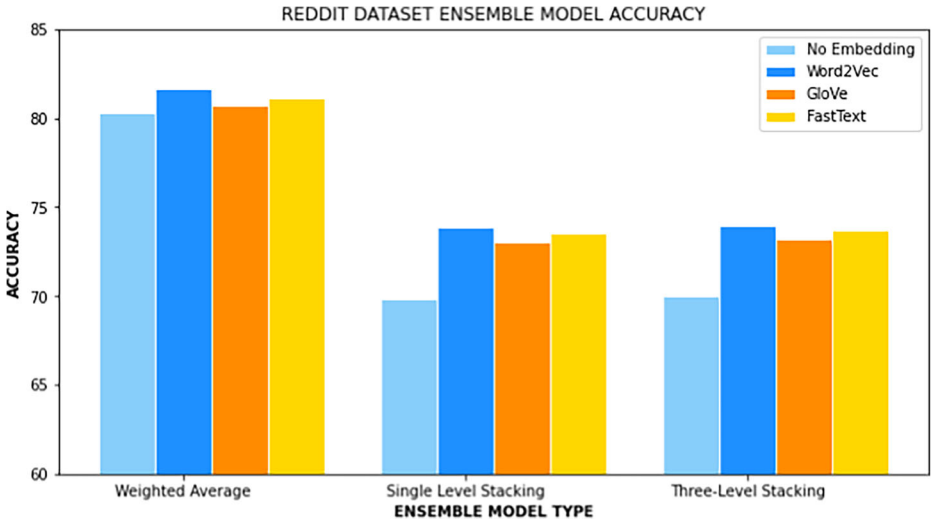

Fig. 21 Reddit dataset ensemble model accuracy

model performs finer when compared to CNN, LSTM and GRU classifiers implemented separately. The reason for the enhanced accuracy in our proposed model is the use of ensemble models such as weighted average ensemble model and level-based stacking ensemble model. Our proposed model combines the strengths of three deep learning models, CNN, GRU and LSTM techniques and is also able to overcome the limitations such as overfitting and underfitting of the stated individual models which can also be a reason for the outstanding performance of the model. Hence, instead of using a single deep learning architecture, CNN, GRU and LSTM models are combined to cover the problems so as to improvise the stability, accuracy and also the predictive power of the proposed model. Using a weighted average ensemble has further helped in improving the accuracy of our model because it allows the contribution of each ensemble member to a prediction to be weighted proportionally to the trust or performance of the member on a holdout dataset.

Some errors that we have faced during our study:

- False negatives: sarcastic tweets not being detected by the model, most probably because they are very specific to a certain situation or culture and they require a high level of world knowledge that Deep Learning models don't have. The most effective sarcasm is the one tailored specifically to the person, situation and relationship between the speakers.
- Sarcastic tweets written in a very polite way are undetected. Sometimes people use politeness as a way of being sarcastic, highly formal words that don't match the casual conversation. Complimenting someone in a very formal way is a common way of being sarcastic.

5 Conclusion and future scope

This study aims to bridge the gap between human and machine intelligence to enable the latter to recognize and understand sarcastic behavior and patterns. The proposed ensemble model using Baseline CNN, Bi-Directional LSTM and GRU may be used to accomplish this task. In

order to improve the accuracy of the proposed model, the required dataset is prepared on different previous trained word-embedding models and hence their accuracies are compared.

There is a significant improvement in employing word embeddings as compared to the ensemble accuracies without these embeddings, the study concluded that the best performing model was LSTM using GloVe embeddings with obtained accuracy as 95.36%, 96.10% and 95.64% respectively for CNN, LSTM and GRU model while with ensemble model, the weighted average with Glove word embedding came out to be 98.97%. For the Reddit dataset, Word2Vec performed the best with accuracy obtained as 73.19%, 73.65% and 73.34% for CNN, LSTM and GRU model while the weighted average for the ensemble model came out to be 81.64%. The weighted averages for the ensemble model without word embeddings came out to be 98.09% and 80.27% for both the datasets.

Based on the experimental results, it tends to be concluded that for both the datasets, the LSTM technique has performed the best among all the models and also the weighted average ensemble model has achieved more accuracy when compared to other models.

This model can be expanded further by making use of different techniques like ELMo and BERT. Hyperparameter tuning can further improve the model's characteristics. The future scope of this study is to build upon the existing model and use the results of the above research as a baseline moving forward. New challenges have been presented to the existing framework hence, the need is to construct a system dynamic and robust enough to adjust to the circumstances in real time and detect the existence of sarcasm in textual information.

Declarations

Conflicts of interests/competing interests The authors declare there is no conflicts of interests / competing interests at all associated with this manuscript.

References

1. Al-Moslmi T, Omar N, Abdullah S, Albared M (2017) Approaches to cross-domain sentiment analysis: a systematic literature review. Ieee access 5:16173–16192
2. Aloufi S, El Saddik A (2018) Sentiment identification in football-specific tweets. IEEE Access 6:78609–78621
3. Alzubi, J., Nayyar, A., & Kumar, A. (2018, November). Machine learning from theory to algorithms: an overview. In journal of physics: conference series (Vol. 1142, no. 1, p. 012012). IOP publishing.
4. Amir, S., Wallace, B. C., Lyu, H., Carvalho, P., & Silva, M. J. (2016, August). Modelling context with user Embeddings for sarcasm detection in social media. In proceedings of the 20th SIGNLL conference on computational natural language learning (pp. 167-177).
5. Bakshi, R. K., Kaur, N., Kaur, R., & Kaur, G. (2016, March). Opinion mining and sentiment analysis. In 2016 3rd international conference on computing for sustainable global development (INDIACom) (pp. 452-455). IEEE.
6. Barbieri, F., Saggion, H., & Ronzano, F. (2014, June). Modelling sarcasm in twitter, a novel approach. In proceedings of the 5th workshop on computational approaches to subjectivity, sentiment and social media analysis (pp. 50-58).
7. Bark, O., Grigoriadis, A., Pettersson, J., Risne, V., Siitova, A., & Yang, H. (2017). A deep learning approach for identifying sarcasm in text (Bachelor's thesis).
8. Bharti SK, Vachha B, Pradhan RK, Babu KS, Jena SK (2016) Sarcastic sentiment detection in tweets streamed in real time: a big data approach. Digital Communications and Networks 2(3):108–121
9. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5:135–146
10. Bouazizi M, Ohtsuki T (2017) A pattern-based approach for multi-class sentiment analysis in twitter. IEEE Access 5:20617–20639

11. Bouazizi M, Ohtsuki T (2018) Multi-class sentiment analysis in twitter: what if classification is not the answer. IEEE Access 6:64486–64502
12. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.
13. Dave, K., Lawrence, S., & Pennock, D. M. (2003, May). Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In proceedings of the 12th international conference on world wide web (pp. 519-528).
14. Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., & Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. Stat, 1050, 1.
15. Fersini, E., Pozzi, F. A., & Messina, E. (2015, October). Detecting irony and sarcasm in microblogs: the role of expressive signals and ensemble classifiers. In 2015 IEEE international conference on data science and advanced analytics (DSAA) (pp. 1-8). IEEE.
16. Filatova, E. (2012, May). Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In Lrec (pp. 392-398).
17. Ghosh, A., & Veale, T. (2016, June). Fracking sarcasm using neural network. In proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis (pp. 161-169).
18. Ghosh, D., Fabbri, A. R., & Muresan, S. (2017). The role of conversation context for sarcasm detection in online interactions. arXiv preprint arXiv:1707.06226.
19. Hazarika, D., Poria, S., Gorantla, S., Cambria, E., Zimmermann, R., & Mihalcea, R. (2018). Cascade: contextual sarcasm detection in online discussion forums. arXiv preprint arXiv:1805.06413.
20. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
21. Jain D, Kumar A, Garg G (2020) Sarcasm detection in mash-up language using soft attention based bi-directional LSTM and feature-rich CNN Applied Soft Computing:106198
22. Joshi, A., Tripathi, V., Patel, K., Bhattacharyya, P., & Carman, M. (2016, November). Are word embedding-based features useful for sarcasm detection?. In proceedings of the 2016 conference on empirical methods in natural language processing (pp. 1006-1011).
23. Joshi A, Bhattacharyya P, Carman MJ (2017) Automatic sarcasm detection: a survey. ACM Computing Surveys (CSUR) 50(5):1–22
24. Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). fastText. Zip: compressing text classification models. arXiv preprint arXiv:1612.03651.
25. Khodak, M., Saunshi, N., & Vodrahalli, K. (2018, May). A large self-annotated Corpus for sarcasm. In proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).
26. Kreuz RJ, Roberts RM (1995) Two cues for verbal irony: hyperbole and the ironic tone of voice. Metaphor Symb 10(1):21–31
27. Kumar, A., & Garg, G. (2019). Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets. Journal of ambient intelligence and humanized computing, 1-16.
28. Kumar, A., & Jaiswal, A. (2017). Empirical study of twitter and tumblr for sentiment analysis using soft computing techniques. In proceedings of the world congress on engineering and computer science (Vol. 1, pp. 1-5).
29. Kumar A, Jaiswal A (2020) Systematic literature review of sentiment analysis on twitter using soft computing techniques. Concurrency and Computation: Practice and Experience 32(1):e5107
30. Kumar A, Sebastian TM (2012) Sentiment analysis on twitter. International Journal of Computer Science Issues (IJCSI) 9(4):372
31. Kumar A, Teeja MS (2012) Sentiment analysis: a perspective on its past, present and future. International Journal of Intelligent Systems and Applications 4(10):1–14
32. Kumar A, Sangwan SR, Arora A, Nayyar A, Abdel-Basset M (2019) Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. IEEE Access 7:23319–23328
33. Kumar A, Sangwan SR, Arora A, Nayyar A, Abdel-Basset M (2019) Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. IEEE access 7:23319–23328
34. Kumar A, Narapareddy VT, Srikanth VA, Malapati A, Neti LBM (2020) Sarcasm detection using multi-head attention based bidirectional LSTM. IEEE Access 8:6388–6397
35. Kumar A, Sangwan SR, Nayyar A (2020) Multimedia Social Big Data: Mining. In: Multimedia social big data: Mining, In multimedia big data computing for IoT applications (pp. 289–321). Springer, Singapore
36. Kumari A, Behera RK, Sahoo KS, Nayyar A, Kumar Luhach A, Prakash Sahoo S (2020) Supervised link prediction using structured-based feature extraction in social network. Practice and Experience, Concurrency and Computation, p e5839
37. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324

38. Lemmens, J., Burtenshaw, B., Lotfi, E., Markov, I., & Daelemans, W. (2020, July). Sarcasm detection using an ensemble approach. In proceedings of the second workshop on figurative language processing (pp. 264-269).
39. Ling, J., & Klinger, R. (2016, May). An empirical, quantitative analysis of the differences between sarcasm and irony. In European semantic web conference (pp. 203-216). Springer, Cham.
40. Liu, P., Chen, W., Ou, G., Wang, T., Yang, D., & Lei, K. (2014, June). Sarcasm detection in social media based on imbalanced classification. In international conference on web-age information management (pp. 459-471). Springer, Cham.
41. Majumder N, Poria S, Peng H, Chhaya N, Cambria E, Gelbukh A (2019) Sentiment and sarcasm classification with multi-task learning. IEEE Intell Syst 34(3):38–43
42. Manohar, M. Y., & Kulkarni, P. (2017, June). Improvement sarcasm analysis using NLP and corpus based approach. In 2017 international conference on intelligent computing and control systems (ICICCS) (pp. 618-622). IEEE.
43. Maynard, D. G., & Greenwood, M. A. (2014, March). Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In LREC 2014 Proceedings. ELRA.
44. Mehndiratta P, Soni D (2019) Identification of sarcasm in textual data: a comparative study. Journal of Data and Information Science 4(4):56–83
45. Mehndiratta P, Soni D (2019) Identification of sarcasm using word embeddings and hyperparameters tuning. J Discret Math Sci Cryptogr 22(4):465–489
46. Mehndiratta P, Sachdeva S, Soni D (2017) Detection of sarcasm in text data using deep convolutional neural networks. Scalable Computing: Practice and Experience 18(3):219–228
47. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
48. Mishra, A., Dey, K., & Bhattacharyya, P. (2017, July). Learning cognitive features from gaze data for sentiment and sarcasm classification using convolutional neural network. In proceedings of the 55th annual meeting of the Association for Computational Linguistics (volume 1: long papers) (pp. 377-387).
49. Misra, R., & Arora, P. (2019). Sarcasm detection using hybrid neural network. arXiv preprint arXiv: 1908.07414.
50. Onan, A. (2019, April). Topic-enriched word embeddings for sarcasm identification. In computer science on-line conference (pp. 293-304). Springer, Cham.
51. Pai PF, Liu CH (2018) Predicting vehicle sales by sentiment analysis of twitter data and stock market values. IEEE Access 6:57655–57662
52. Patro, J., Bansal, S., & Mukherjee, A. (2019, November). A deep-learning framework to detect sarcasm targets. In proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP) (pp. 6337-6343).
53. Pelser, D., & Murrell, H. (2019). Deep and dense sarcasm detection. arXiv preprint arXiv:1911.07474.
54. Pennington, J., Socher, R., & Manning, C. D. (2014, October). GloVe: global vectors for word representation. In proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).
55. Poria S, Cambria E, Hazarika D, Vij P (2016, December) A deeper look into sarcastic tweets using deep convolutional neural networks. In proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers (pp. 1601-1612)
56. Porwal S, Ostwal G, Phadtare A, Pandey M, Marathe MV (2018, June) Sarcasm detection using recurrent neural network. In 2018 second international conference on intelligent computing and control systems (ICICCS) (pp. 746-748). IEEE
57. Potamias RA, Siolas G, Stafylopatis AG (2020) A transformer-based approach to irony and sarcasm detection. Neural computing and applications, 1-12
58. Saha S, Yadav J, Ranjan P (2017) Proposed approach for sarcasm detection in twitter. Indian J Sci Technol 10(25):1–8
59. Sarsam SM, Al-Samarraie H, Alzahrani AI, Wright B (2020) Sarcasm detection using machine learning algorithms in twitter: a systematic review. Int J Mark Res 62(5):578–598
60. Shayaa S, Jaafar NI, Bahri S, Sulaiman A, Wai PS, Chung YW, … Al-Garadi MA (2018) Sentiment analysis of big data: methods, applications, and open challenges. IEEE Access 6:37807–37827
61. Sobti P, Nayyar A, Nagrath P (2021) EnsemV3X: a novel ensembled deep learning architecture for multi-label scene classification. PeerJ Computer Science 7:e557
62. Tarigan J, Girsang G (2018) Word similarity score as augmented feature in sarcasm detection using deep learning. International Journal of Advanced Computer Research. 8. 354–363
63. Tseng CW, Chou JJ, Tsai YC (2018) Text mining analysis of teaching evaluation questionnaires for the selection of outstanding teaching faculty members. IEEE Access 6:72870–72879

64. Wang K, Bansal M, Frahm JM (2018, March) Retweet wars: tweet popularity prediction via dynamic multimodal regression. In 2018 IEEE winter conference on applications of computer vision (WACV) (pp. 1842-1851). IEEE
65. Wu D, Chi M (2017) Long short-term memory with quadratic connections in recursive neural networks for representing compositional semantics. IEEE Access 5:16077–16083
66. Zhang M, Zhang Y, Fu G (2016, December) Tweet sarcasm detection using deep neural network. In proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers (pp. 2449-2460)