Arabian Journal for Science and Engineering (2022) 47:2499–2511
https://doi.org/10.1007/s13369-021-06227-w

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

# Improving Sentiment Analysis for Social Media Applications Using an Ensemble Deep Learning Language Model

**Ahmed Alsayat**[1]

Received: 5 January 2021 / Accepted: 12 September 2021 / Published online: 11 October 2021


**Abstract**
As data grow rapidly on social media by users' contributions, specially with the recent coronavirus pandemic, the need to acquire knowledge of their behaviors is in high demand. The opinions behind posts on the pandemic are the scope of the tested dataset in this study. Finding the most suitable classification algorithms for this kind of data is challenging. Within this context, models of deep learning for sentiment analysis can introduce detailed representation capabilities and enhanced performance compared to existing feature-based techniques. In this paper, we focus on enhancing the performance of sentiment classification using a customized deep learning model with an advanced word embedding technique and create a long short-term memory (LSTM) network. Furthermore, we propose an ensemble model that combines our baseline classifier with other state-of-the-art classifiers used for sentiment analysis. The contributions of this paper are twofold. (1) We establish a robust framework based on word embedding and an LSTM network that learns the contextual relations among words and understands unseen or rare words in relatively emerging situations such as the coronavirus pandemic by recognizing suffixes and prefixes from training data. (2) We capture and utilize the significant differences in state-of-the-art methods by proposing a hybrid ensemble model for sentiment analysis. We conduct several experiments using our own Twitter coronavirus hashtag dataset as well as public review datasets from Amazon and Yelp. For concluding results, a statistical study is carried out indicating that the performance of these proposed models surpasses other models in terms of classification accuracy.

**Keywords** Machine learning · Deep learning · Sentiment analysis · Data mining · Ensemble algorithms · Social media · Pandemic · Coronavirus · COVID-19

## 1 Introduction

The rise of Internet technology has played an unprecedented role in increasing the number of social media and e-commerce platforms. In addition, users are now accustomed to the idea of expressing their feelings and emotions with others by using these platforms either by text or multimedia data [1–4]. This phenomenon has resulted in the production and generation of a large variety of data, which can be analyzed for assessing sentiment. It is beneficial for individuals and organizations to analyze sentiment, especially given this immense production of data [5]. However, as noted in [6], the identification, continuous monitoring, and filtering of the information present on social media applications to analyze sentiment are challenging. Some of the factors are the presence of unstructured data, differences in languages, diversity of websites and social media platforms, and heterogeneous data about the opinions of individuals. Therefore, appropriate tools and algorithms are required to analyze the sentiment from the data that are gathered from social media in big data, fog computing blockchain, and IoT-based platforms [7,8].

Sentiment analysis involves examining the opinions, perceptions, attitudes, thoughts and emotions of individuals shared on different social media platforms. In particular, sentiment analysis aims to classify a particular written text as a neutral, positive, or negative sentiment [9,10]. Dang et al. [11] identified three main approaches in sentiment analysis: machine learning based, lexicon based, and hybrid. The lexicon-based technique is categorized into two approaches: corpus and dictionary based. In a dictionary-based approach, the classification of sentiment is carried out by utilizing a dictionary of terms such as those found in WordNet and Sen-

✉ Ahmed Alsayat
asayat@ju.edu.sa

[1] Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakaka 72388, Kingdom of Saudi Arabia

tiWordNet. In contrast, the corpus-based analysis approach depends on the content's statistical analysis by utilizing techniques associated with the k-nearest neighbors, hidden Markov models (HMM), and conditional random field (CRF). Thus, the corpus-based approach is not dependent on a predefined dictionary. In addition, machine-learning-based sentimental analysis techniques involve traditional models and deep learning models. However, the hybrid approach combines machine learning and lexicon approaches. These traditional approaches of sentiment analysis have gained popularity due to their effective results; however, one of the drawbacks of these approaches is that they incorporate feature engineering. Thus, the deep learning approach was introduced because of its capability of learning written text without the need to perform manual feature engineering, hence outperforming other methods of sentiment analysis. Araque et al. [12] stated that the underlying idea behind using a deep learning approach for analyzing sentiment is to learn about new complex features, which are extracted from the data without any additional or extensive contributions. The algorithms of deep learning are efficient, as they do not request any manually crafted features as input; rather, they generate complex features through self-learning. However, it is significant to note that deep learning algorithms demand an immense amount of data to act and work effectively.

The recent cloud platforms provided by IBM, Amazon, Google, and Microsoft show high performance in terms of sentiment analysis accuracy. The study in [13] demonstrated how these off-shelf-technologies outperform the bag-of-words approach by analyzing a social media dataset. The output of the experiment shows that IBM Watson Natural Language Understanding archived the highest accuracy compared to other platforms and gained more than 30% accuracy compared to the bag-of-words approach, which has the lowest accuracy between them. This indicates that cloud platforms have high performance, and accuracy when working with text analysis.

According to [14,15], deep learning involves applying artificial neural networks to learn different tasks using networks that are attributed to different layers. The search primarily takes inspiration from the way that the human brain is structured, as it contains a large number of entities (neurons) that are used for processing the information. This is mainly categorized into feedforward and recursive neural networks. The use of neural networks plays an important role at different levels for analyzing sentiment, including the document level, aspect level, and sentence level. In sentence-level analysis, it is determined whether each sentence is an opinion; however, in document-level analysis, the opinion of the entire document is determined. However, at the aspect level, a detailed analysis is undertaken that mainly uses the natural language processing technique [16]. Habimana et al. [17] identified different deep learning approaches that have been extensively used for analyzing sentiment and prominently include deep reinforcement learning, recurrent neural networks, convolutional neural networks, unsupervised pretrained networks, and hybrid deep learning neural networks. Based on these findings, numerous studies have been conducted that utilized artificial intelligence and deep learning technology for adequately conducting sentiment analysis.

Reference [12] noted that the ensemble approach incorporates a set of models that are particularly classifiers for generating a new model and is more efficient and reliable than a single model. Some prominent ensemble techniques include boosting and bootstrap aggregating, i.e., bagging, and the random subspace method [18]. Only a few researchers have explored the impact of using an ensemble approach for sentiment analysis since the ensemble approach can offer greater accuracy than a single model. Among those studies,[19] utilized the ensemble approach to analyze sentiment in English language tweets. In this regard, the researcher used deep learning techniques of convolutional neural networks and long short-term memory (LSTM). Behera et al. [20] also proposed a convolutional LSTM model for sentiment analysis in social big data, and Christos L. Stergiou et al. [21] proposed a model that offers users a safer and efficient environment for browsing the Internet and sharing and managing large-scale data in the fog.

However, the authors in [22] proposed a model by using recurrent and neural networks in combination to analyze the sentiment of short texts extracted from social media platforms. In this account, local features were extracted by using a convolutional neural network (CNN), and recurrent neural networks were utilized for learning the long-distance dependencies that also aided in sentence-level feature representation. This combination eventually provided higher classification accuracy than the preexisting models of LSTM and gated recurrent units (GRUs) on the three corpora with 82.28%, 51.50%, and 89.95% accuracy. Similarly,[23,24] proposed an ensemble approach by combining 10 LSTMs and 10 CNNs using the soft voting approach for analyzing sentiment from a Twitter dataset in the English language. The imbalanced dataset was treated by utilizing the cross-entropy as a loss function, which was implemented in TensorFlow. The experiment produced favorable and higher results on the five English subtasks using the performance metric of accuracy and F-measure. In addition,[12] used the ensemble approach for the surface and deep features along with the classifiers, where six public datasets of Twitter were used to analyze movie reviews. However, a literature gap is present related to the use of ensemble learning for analyzing sentiment from social media applications. Therefore, the present research study aims to contribute to the literature by using the ensemble approach, where different deep learning models are combined to analyze the sentiment extracted from the data of social media applications. The performance of different

classifiers used in the ensemble approach is also compared with certain performance metrics, resulting in identifying the best deep learning approach. This approach can be used to accurately analyze sentiment on different social media platforms.

In this paper, we propose an enhanced ensemble deep learning model to tackle sentiment analysis tasks. Multiple contributions are provided by our work, including (1) a deep learning framework based on the FastText word embedding technique [25] and an LSTM network that captures contextual relations among words and understands unseen or rare words by recognizing suffixes and prefixes using training data, (2) designing and conducting a statistical experiment to assess significant differences between state-of-the-art methods and proposing a hybrid ensemble model for sentiment analysis, and (3) creating a data extraction pipeline to collect and tag novel coronavirus pandemic data from Twitter that can also be used for any other emerging situations. We also use social media datasets other than the COVID dataset to evaluate the performance of our proposed framework. The rest of this paper is structured as follows. In Sect. 2, we illustrate an intensive study of recent related works regarding sentiment classification using different methods. In Sect. 3, we present the methodology of some related algorithms and present the proposed ensemble deep learning models from previous work. In Sect. 4, we evaluate the experiment and analysis by applying the ensemble deep learning model to social media datasets according to the user's perspective of coronavirus and use other datasets for comparison. The results and discussion are presented in Sect. 5, while the conclusion of the work is presented in Sect. 6.

## 2 Related Works

### 2.1 Sentiment Analysis and Its Application

Sentiment analysis involves investigating the approach of a writer toward a particular subject or the overall contextual polarity of an entire document [26]. The underlying purpose of sentiment analysis is to classify texts based on sentiment or opinion, not by topic [27]. In particular, sentiment analysis incorporates the use of information retrieval, NLP, data mining, and knowledge management techniques for identifying and extracting subjective information from a large volume of unstructured data [28]. As per [29], sentiment analysis is a complex process that includes five phases for analyzing the sentiment in the source materials. These phases include the collection of data, preparation of text, detection of sentiment, classification of sentiment, and presentation of findings. The sentiment analysis technique is applied mainly in two approaches: supervised learning and unsupervised learning [30]. The supervised learning approach involves sorting the training set to create text-based patterns. The unsupervised learning approach does not involve the use of a database but rather is based on the set of words where the terms *negative* and *positive* are considered. Therefore, the frequency in terms of the negative and positive in the entire text provides an indication for tagging the document based on these terms [31,32].

Sentiment analysis is used in several diverse fields. The authors in [33] stated that sentiment analysis assists the government in identifying their strengths and weaknesses by examining public opinions on social media platforms. Likewise, in online commerce, sentiment analysis is performed to convert dissatisfied customers into promoters by analyzing their shopping experience and opinions regarding product quality [34]. Vohra & Teraiya [35] affirmed that sentiment analysis is used for assessing customer reviews and opinions about products and services. Tweetfeel is an exemplary application that analyzes tweets in a real-time manner [36]. Wang et al. [37] also highlighted the application of sentiment analysis in Blogger-centric contextual advertising, which involves developing personal advertisements on blog pages according to the interests of the brands. Based on these findings, sentiment analysis is widely implemented in different fields for identifying and assessing particular behavioral patterns and sentiment.

### 2.2 Deep Learning Approaches in Text Classification

Deep learning approaches have gained immense popularity over machine learning algorithms [38–40]. This is because deep learning approaches provide reliable results regarding text classification. Their success is mainly credited to a capacity to model nonlinear and complex relationships within the data [41,42]. There are three main types of deep learning approaches that are used for classifying text and documents: deep neural networks (DNNs), recurrent neural networks (RNNs), and CNNs.

Anqi et al. [43] applied the RNN technique to predict the citation count for journal papers in the field of artificial intelligence. To predict the citation count, the experiment specifically implemented bidirectional LSTM on paper metadata text. The study shows good performance in terms of predicting the count citation of a paper.

Mittal et al. [44] proposed deep graph-LSTM for text classification. The model used the graph database to store its documents. The experiment was verified on legal cases of the Indian judiciary. The study produced an accuracy of 99% when classifying the related category of a fresh case.

Deepika et al. [45] proposed a model of accelerated gradient LSTM where the Kalman filter is applied to reduce the noise and errors of data. The study was applied to predict the stock market where the data were collected from Twitter and

Yahoo. The model achieved better performance when using the Kalman filter, reaching accuracy of 90.42%.

Hasni et al. [46] proposed a deep learning model that used neural networks to locate infected area of COVID-19. The model was applied to tweets, which were collected from the UK and the USA. The experiment revealed that using bidirectional LSTM increases the accuracy of geolocation.

## 2.3 Ensemble Methods in Sentiment Analysis Guidelines

The study in [10] performed sentiment analysis on Arabic language tweets. In this account, the search proposed learning sentiment-specific word embeddings for the classification of Arabic tweets. In that study, three datasets of Arabic tweets were used. In particular, the sentiment classifier of support vector machine (SVM) with LibLearner was used to classify the tweets as positive or negative. The experimental study used baseline and surface features through an ensemble approach and Collobert and Weston (C&W), Arabic sentiment embeddings constructed using the prediction (ASEP), Arabic sentiment embeddings constructed using ranking (ASER), Arabic sentiment embeddings constructed using hybrid (ASEH), and bidirectional encoder representations from transformers (BERT) models [47]. To examine the effectiveness of pooling functions, the max, min, average, and concatenation pooling functions were used, which showed that the average function provided the highest performance over most of the models. The study found that the use of the ensemble approach for the deep learning models provided the highest F1 score, i.e., 80.38% on the dataset of Arabic tweets with surface features and generic embeddings. Another study, conducted by Heikal et al. [22], used the ensemble method, which combines deep learning models, i.e., LSTM and CNN models, to analyze the sentiment in Arabic tweets. The ensemble model utilized the soft voting technique, whose performance was evaluated using the F1 score performance metric. The use of the ensemble technique produced a 64.46% F1 score, which outperformed individual deep learning models.

To analyze the financial sentiment, the authors in [48] proposed an ensemble model that combined classic feature-based and deep learning models by utilizing the multilayer perceptron (MLP) network. The MLP network contained two hidden layers, where each layer had four neurons and the last layer used ReLU activation and tanh activation functions. The evaluation was performed using the cosine function and showed that the ensemble approach yielded higher cosine scores of 0.797 and 0.786. The authors in [28] also proposed an ensemble deep learning model that combines a character-level CNN and word-level CNN on Twitter's dataset for detecting drug abuse behavior. The proposed model classifies the dataset into positive and negative tweets, where the extracted features from the tweets through word-level CNN and character-level CNN are forwarded to the meta-learner that provides the final predictions. The results from the ensemble deep learning model were compared with the ensemble machine learning model and showed that in a highly imbalanced dataset with a 30:70 split, the ensemble deep learning model provides better results when using the F1 score as a performance metric. Likewise,[49,50] proposed an ensemble learning approach by introducing an SVM classifier with a CNN for analyzing the sentiment in data collected from microblogs and other social media sites. The researcher used a crawler for crawling data from microblogs, which was then treated through a corpus and fed as an input sample of CNN to develop a classifier based on SVM/RNN. The results showed that the solution can implement embeddings constructed using the prediction (ASEP), and commendably improve the accuracy of emotional orientation. Therefore, it can be affirmed that CNN-based classifiers have a greater tendency to accurately analyze the sentiment from the data that are extracted from social media platforms.

Moreover, sentiment analysis can be significantly enhanced using ensemble deep learning models [51]. One such model has been used to examine the sentiment terms using co-extraction that examined the sentiments based on polarity and intensity. Thus, this state-of-the-art method showed higher effectiveness in sentiment analysis than other methods. Furthermore, the authors in [52] used LSTM and CNN models to form a hybrid model for the analysis of movie reviews posted by people. The model used text data analysis to determine the sentiments and emotions of the people who watched the movie. It was noted that the model showed an overall accuracy of 91%. In addition, Al-Makhadmeh & Tolba [53] used the ensemble deep learning approach (KNLPEDNN) to automatically examine hate speech on social media. This model also worked on data collected on various hate speech texts that helped in the identification of hate speech. The study found that the model is highly effective with an accuracy of 98.71%. Therefore, it can be noted from these studies that ensemble deep learning models are highly beneficial and accurate in sentiment analysis on social media platforms. Several popular hybridization methods will be discussed in the following subsections:

### 2.3.1 Bagging

Bagging is a popular hybridization method that is used in ensemble deep learning to use more than one model. As per [54], bagging involves the reduction of variance by generating extra data to train datasets using various combinations. Through this practice, multi-set data are produced that enable the hybridization of more than one deep learning model. The study in [55] used a bagging-based method along with naïve Bayes trees to create a hybrid model for estimation. It was

noted during the study that the bagging method is usually common for estimation techniques by studying the data. For sentiment analysis, bagging is used to combine more than one model so that their benefits can be fully exploited. Thus, this method of hybridization is convenient due to its simplicity, which makes hybridization more efficient.

#### 2.3.2 Boosting

Boosting is another key hybridization technique that is used to combine more than one deep learning model. According to [56], boosting an iterative technique works by adjusting the weight of observations based on their last classification. This method considers homogenous weak learners by learning them sequentially and combines them using a deterministic approach. Ardabili et al. [57] used boosting to combine the decision tree model and regression tree model. The iterative technique used by boosting is key to the combination of the models as it helps reduce the bias and variance between different models. In sentimental analysis, boosting is an effective method of hybridization to systematically study the data and combine more than one model to perform an accurate operation. Thus, boosting is very beneficial due to its deterministic and iterative approach that ensures high accuracy.

#### 2.3.3 Stacking

Stacking is a hybridization technique that works using a parallel approach to data training. The data are trained in parallel from different models to produce a meta-model that has a very low bias. This is a major advantage of this technique, as it ensures higher accuracy. However, it might not be very effective to reduce the variance among the component models that can make the results less reliable. The study in [56] highlighted that stacking is an efficient method of ensemble deep learning as it saves substantial training time due to the parallel training approach. Nonetheless, the use of this technique is limited because stacking can cause a problem in sentimental analysis that must work through a large volume of data, which can make stacking less efficient [54]. Thus, the use of the hybridization technique is based on the needs of data training and analysis.

After deep research in previous studies related to ensemble methods for sentiment analysis, we conclude that the ensemble technique is more efficient for sentiment analysis. Moreover, we find that our proposed ensemble model is considered a unique model that combines state-of-the-art methods and uses a deep learning framework based on the FastText word embedding technique and an LSTM network.

## 3 Methods

In this section, our method is presented to predict the sentiment of a given text using customized ensemble deep learning methodology. The customized model is based on the advanced FastText word embedding technique [25] for representing feature space and LSTM networks [58], which are special kinds of RNNs. RNNs are good for modeling languages because language is a sequence of words and each word shares semantic meaning with the words next to it. Furthermore, LSTM networks are capable of remembering long-term dependencies and enhance the efficiency of RNNs. We refer to our approach as the customized ensemble deep learning language model. We start by reviewing the basic functions of LSTMs and word embedding techniques, and then discuss the detailed implementation of the proposed algorithm. Regarding the feature space, the FastText method allows our proposed model to capture the meaning of shorter words and allows the model to understand unseen words by recognizing suffixes and prefixes using an embedding technique.

### 3.1 Word Embedding

The bag-of-words model appears to be very high dimensional in general terms because of the existing lack of contextual relations between words. To better represent the limited content in short texts, especially when working with pandemic content where new words and terminologies are used, we use an advanced word embedding model [25] to learn the contextual relations among words and to understand unseen or rare words by recognizing suffixes and prefixes in training data. In this model, the representation of each word is formed as a bag of character n-grams in addition to the word itself. For example, the word "matter," with $n = 3$ generates the representation for the character $n$-grams as <ma, mat, att, tte, ter, er>. To differentiate the $n$-grams of a word from the original form of a word itself, brackets are added in this case as boundary symbols. Therefore, if the vocabulary contains any parts of the word "mat," that vocabulary is represented as <mat>. This scenario assists in maintaining and preserving the meaning of shorter words that may appear as n-grams of other words. Furthermore, this allows for the inherent capture of meaning for suffixes and prefixes [59].

### 3.2 Long Short-Term Memory (LSTM)

After representing each word by its corresponding feature vector representation using the word embedded model, the feature set is input to the LSTM network in sequence form. The capability of learning long-term dependencies between input features is an aspect of LSTM, which is a special type of RNN. A chain of repeating modules is a special form of

all RNNs and is considered a simple structure in the standard of all RNNs. This repeating module works in the opposite manner when working with LSTM, where it is more complicated. Rather than the singularity of the layer contained in neural networks, there are four layers existed (forget gates, input gates, new memory gates and output gates), and all act in a special manner [60].

LSTM consists of two states: hidden state and cell state. At a particular time step $t$, LSTM decides which information must be taken from the state of the cell. The decision is made by a sigmoid function layer $\sigma$ called the forget gate. The function takes $h_{t-1}$ (output from the previous hidden layer) and $x_t$ (current input) and outputs a number in [0, 1]. In this case, 1 represents "completely keeping in," and 0 represents "completely taken away" in the equation below.

$$f_t = \sigma(W^f x_t + U^f h_{t-1}). \tag{1}$$

The LSTM then determines what new information to keep in the cell state. There are two steps. The first step interacts with the "input gate" as in Eq. 1, where this gate is a sigmoid function layer. The duty of this function is to specify an LSTM in which the values are updated. Second, a vector of new candidate values $\tilde{C}$ is created by the tanh function layer. This step adds the state of the cell. These steps are combined by LSTM to start creating an update to the state.

$$i_t = \sigma(W^i x_t + U^i h_{t-1}) \tag{2}$$

$$\tilde{C}_t = \tanh(W^n x_t + U^n h_{t-1}). \tag{3}$$

At this point, the model updates the old cell state $C_{t-1}$ into a new cell state $C_t$ as represented in Eq. 4. Notably, the gradient can be controlled when going across the forget gate $f_t$ and allows for deletes and updates for explicit "memory." This procedure helps alleviate vanishing gradients or any problems associated with the exploding gradient in the standard RNN.

$$C_t = f_t * Ct - 1 + i_t * \tilde{C}_t. \tag{4}$$

Finally, based on the state of the cell, LSTM determines its output. LSTM first enables a sigmoid layer where it determines which parts of the cell state to transfer as output in Eq. 5, called the "output gate." At this stage, via the function of tanh, LSTM determines the state of the cell and decides the part of output as Eq. 6.

$$o_t = \sigma(W^o x_t + U^o h_{t-1}) \tag{5}$$

$$h_t = o_t * \tanh(C_t). \tag{6}$$

For compatibility with the sequential input of LSTM, we first convert tweets or posts text into a three-dimensional matrix $M(X, Y, Z)$, where $X$ is the feature representation from the word embedding model, $Y$ is the number of words in the text, and $Z$ is the number of tweets or posts. In the input layer, the number of neurons is the same as the dimension of the feature set. The number of neurons in the output layer is the number of classes, which is 2 in our case (positive or negative sentiment). At each point, and by gradient-based back propagation over time, we are able to adjust the weights of edges in the hidden layer. The sentiment classification model can be obtained after several tests and several training epochs.

## 3.3 Other Methods

We also use available NLP libraries from Google, Microsoft, and IBM in our experiments.

### 3.3.1 Google

Google's Cloud Natural Language API provides natural language understanding technology, which includes sentiment analysis, entity analysis, entity sentiment analysis, content classification and syntax analysis. Bidirectional encoder representations (BERT) is the latest NLP algorithm [61] and is a part of the larger cloud machine learning API family from Google.

### 3.3.2 Microsoft

Microsoft Azure is a cloud platform that contains a text analytics API for advanced NLP. This free technology provides its service over raw text with four major functions: sentiment analysis, extraction of key phrases, recognition of named entities, and language detection. Azure cognitive services introduced the API as a part of its family, where in the cloud, a variety of machine learning and AI algorithms are available for any developing projects [13].

### 3.3.3 IBM

IBM Watson is an advanced off-the-shelf technology for artificial intelligent solutions. This free technology runs with the recent worldwide innovation development for machine learning. IBM Watson offers a free API for nature language understanding and performing sentiment analysis as a part of its family. In other words, deep learning on a cloud is developed to explore the knowledge of complex texts for many different classes and levels [62].

## 3.4 Proposed Framework

Figure 1 shows the implementation of the proposed customized ensemble deep learning language framework for sentiment analysis. Tweets are collected as a dataset using

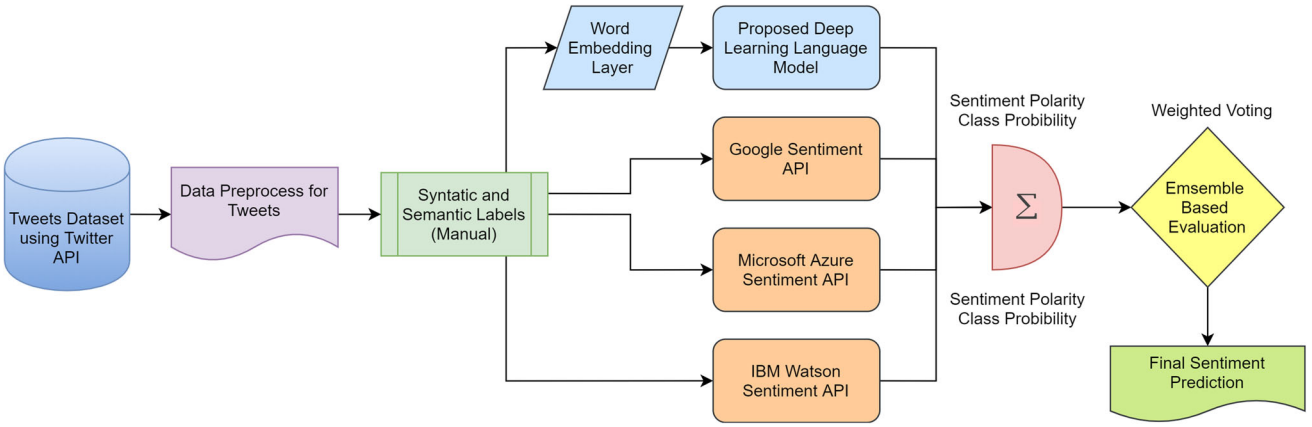



**Fig. 1** Proposed deep learning ensemble model for sentiment analysis

Twitter API. A detailed description of the dataset is presented in Sect. 4.1. These tweets are then processed and manually annotated by the CrowdFlower platform as positive and negative for model training purposes. Once the dataset is cleaned and labeled, we transfer it to different machine learning models for classification. For our proposed model, the dataset first goes through the word embedding layer (described in Sect. 3.1), where the tweet dataset is transformed into a feature set and passes it to a customized deep learning language model. The word embedding layer is based on the FastText word embedding technique [25] and an LSTM network that captures contextual relations among words and understands unseen or rare words by recognizing suffixes and prefixes using training data. During model training, the dataset is divided into training, validation, and test sets to estimate optimal model parameters, especially for classifying short tweets and understanding rare words in the pandemic context. The detailed process of model selection is explained in Sect. 4.3. The same dataset is passed through other models to obtain the respective outcomes. Each output consists of class labels and class probabilities or scores, which represent how strongly each tweet is associated with its predicted label. Before moving to the final ensemble output, a statistical experiment is conducted to assess significant differences between different models. The details for significant testing are explained in Sect. 5.2. In the final stage, all outputs are combined to produce a final sentiment prediction based on ensemble decisions. This ensemble decision is based on the average method, where a majority vote is based on the average of the predicted probabilities. The goal in proposing a customized ensemble model is to improve the overall accuracy by overcoming the shortcomings of weak classifiers. Details of the experimental evaluation are presented in Sect. 4.

## 4 Experimental Evaluation

This section covers the dataset description, model selection and performance evaluation of the proposed model.

### 4.1 Dataset Description

We use 3 different sets of data to evaluate our model and compare it with other models. The first set of data is our custom Twitter covid-19 dataset. This dataset is obtained using the Twitter API with hashtags #COVID-19 and #coronavirus. Once the data are collected, they go through a data preprocessing step for cleaning to remove hashtags and website URLs and links. Since Twitter does not provide any sentiment labels for tweets, manual tagging is performed with the help of the CrowdFlower platform, which is the simplest and most flexible way to scale workforces and accurately complete human evaluation of data and information. CrowdFlower labels each tweet as positive or negative in terms of sentiment. This dataset contains 18,000 total tweets, where 70% of the data are used as the training set and 30% are used for validation and testing. In terms of polarity distribution, this dataset is balanced with 50% positive and 50% negative tweets.

The second set of data consists of two datasets Yelp review and Amazon review data. The Yelp dataset was taken from Yelp dataset Challenge repository in 2015. The polarity level for each review is labeled by considering 1 and 2 stars as negative sentiment, and more than 2 starts as positive sentiment. The entire dataset has approximately 280,000 training records and 19,000 test records in each category of sentiment. The Amazon review dataset is from the Stanford Network Analysis Project(SNAP). It consists of 18 years of data with nearly 34,687,000 reviews from 6,640,000 users on approximately 2,440,000 products [63]. Compared to the Yelp review dataset, this dataset contains approximately 1,800,000 train-

**Table 1** Web 2.0 data description

| Web app | # Records | Pos/Neg distribution |
|---|---|---|
| Twitter | 4242 | 58% / 42% |
| MySpace | 1041 | 85% / 15% |
| YouTube | 3407 | 68% / 32% |
| BBC | 1000 | 14% / 86% |
| Runners World | 1046 | 68% / 32% |
| Digg | 1077 | 27% / 73% |

Total number of records along with the distribution of positive and negative labels for Web 2.0 datasets

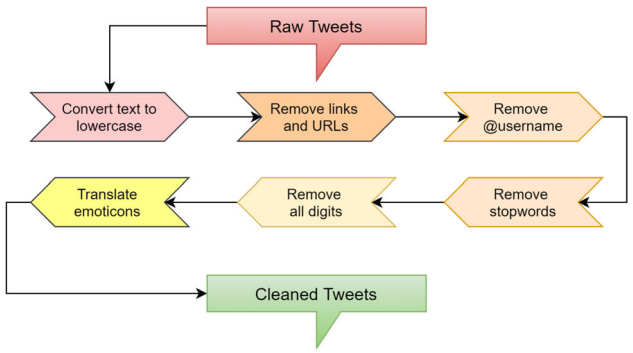



**Fig. 2** Data preprocessing pipeline for our datasets

ing records and 200,000 testing records in each category of sentiment.

The third set of data is referred to as Web 2.0 data that contains labeled messages by humans as positive and negative, and is made available in the SentiStrength search [64]. This set contains six datasets from a wide range of social media applications such as Twitter, MySpace, YouTube, BBC, Runners World and Digg comments. Table 1 provides a summary of each dataset along with the distribution of positive and negative messages.

## 4.2 Data Preprocessing

Data generated by users on social media contain a variety of content other than alphabetic characters such as punctuation, stop words, usernames, graphical icons, web links and URLs. These contents do not contribute to the process of sentiment analysis. For example, the username never supports any algorithm to accurately classify positive or negative tweets. Such content is sometimes referred to as noise, and it is a good practice to remove it to increase the performance of classification algorithms.

Figure 2 shows the data processing stages used during our experiments. In the first stage of this pipeline, all characters in the text are converted into lowercase. Then, all web links and URLs as well as usernames are removed since they do not provide any emotional or sentimental content within the text. Later in this pipeline, we remove punctuation, numbers, and undefined characters. In the last part of data processing, we translate emoticons and graphical icons into positive or negative polarity and use this translation to assign class labels to each tweet. Section 4.2.1 explains the process of emoticon translation.

**Table 2** Translating emoticons and emojis to sentiment polarity

| Icon | Polarity | Symbols |
|---|---|---|
|  | Positive | {:) :] :} :o) :o] :o} :-] :-) :-} =) =] =} =^] =^) =^} :B :-D :-B :^D :^B =B =^B =^D :) :] :} =) =] =} <3 ^-^ :* =* :-* ;) ;] ;} :-p :-P :-b :^p :^P :^b =P =p :P :p :b =b =^p =^P =^b |
|  | Negative | D: D= D-: D^: D^= :( :[ :{ :o( :o[ :^( :^[ :^{ =^( =^{ >=( >=[ >={ >=( >:-{ >:-[ >:-( >=^[ >:-( :-[ :-( =( =[ ={ =^[ >:- =( >=[ >=^( :'( :'[ :'{ ='{ ='( ='[ = : =/ :/ o.O O_o Oo :-{ >:-{ >=^{ :o{ =$ :$ |

Table showing different combinations of characters with their corresponding meanings in terms of emotions, sentiments and polarity

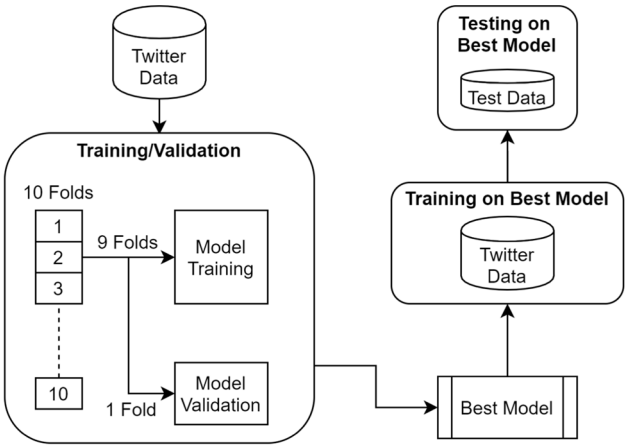
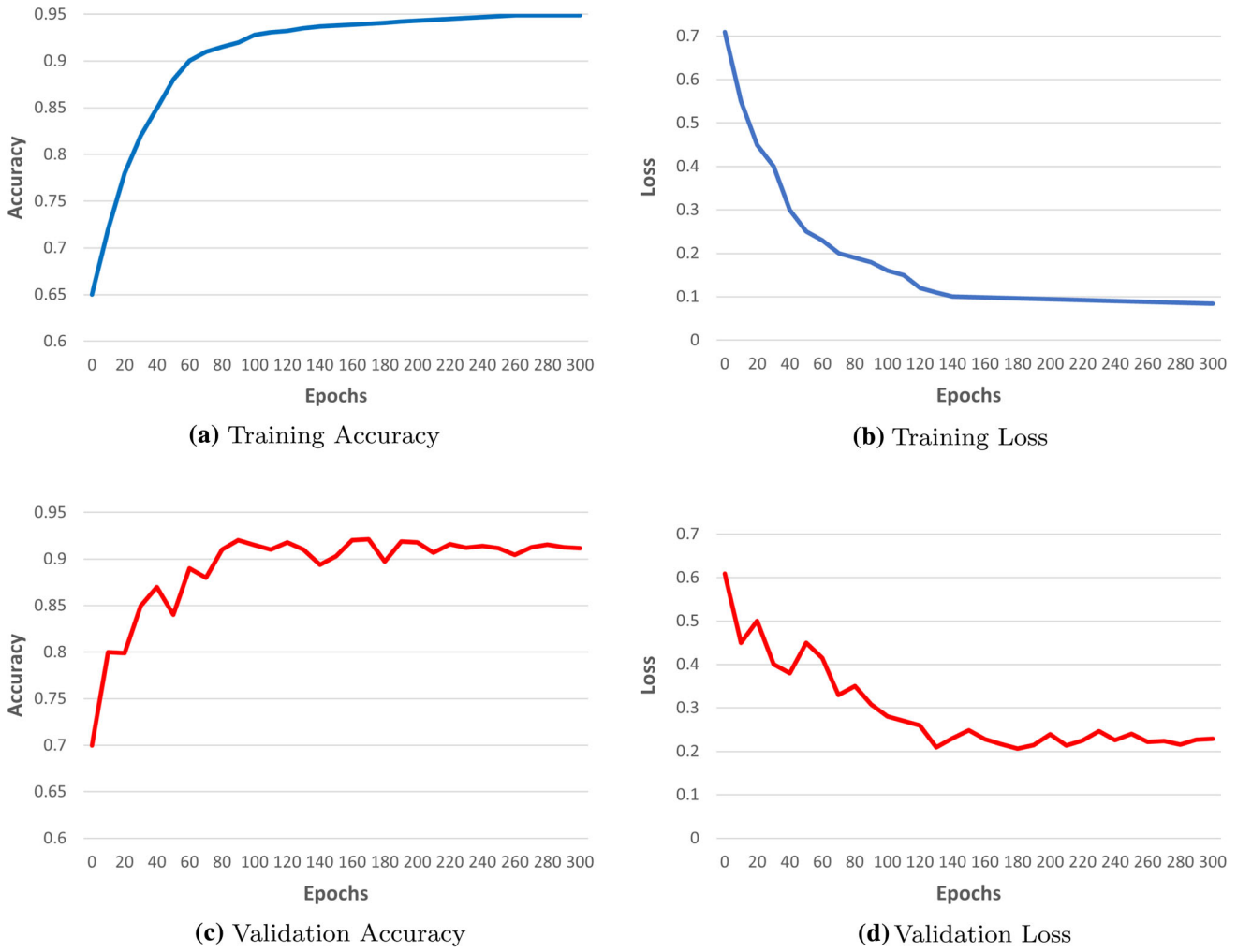



**Fig. 3** Model selection process using different sets of hyperparameters for the proposed deep learning language model

### 4.2.1 Emoticons and Emojis

We also extract the polarity of tweets from emoticons, where we utilize a set of common emoticons as shown in Table 2. We also augment the emoticons by adding different variations of the main primary positive and negative polarities. Text with more than one emoticon is assigned a polarity of the first emoticon that appears in the text to simplify the process.




**(a)** Training Accuracy **(b)** Training Loss

**(c)** Validation Accuracy **(d)** Validation Loss

**Fig. 4** Model evaluation of the proposed deep learning algorithm using accuracy and loss curves during training and validation on the COVID-19 dataset

### 4.3 Model Selection Using Hyperparameter Investigation

The performance and accuracy of our model depend on two different parameters: the total number of hidden neurons and the total number of hidden layers in the network. We use softmax as an activation function on the output layer throughout our experiments. With this scenario, we start with model and parameter selection. To enable this experimental setting, we split our Twitter dataset based on a tenfold cross-validation technique into (i) a training set (ninefold of data out of tenfold), (ii) a validation set (onefold of data out of tenfold excluded from the training set), and (iii) a test set (not included in any of the training and validation sets).

Figure 3 shows the flow diagram of our model selection and evaluation process. After the data split is complete, we train our customized ensemble deep learning language model with a combination of various hyperparameter settings (by using a technique called grid search in data mining). For each set of hyperparameters, we train different models, assess the classification performance on the validation set, and select the model that shows the best accuracy on the validation set. Then, we take the test set as input, feed it into our chosen model, and report the accuracy on this independent test set. The entire process is repeated by using all remaining folds and datasets. Note that there may be a set of parameter combinations that shows better classification performance on the test set than on the validation set. This practice ensures that we are generalizing the model and avoiding overfitting problems in machine learning.

## 5 Results and Discussion

We perform several experiments to evaluate the classification performance of our customized ensemble deep learning

**Table 3** Evaluation of the customized ensemble deep learning language model on the Twitter COVID-19 dataset using different sets of hyperparameters

| # Neurons | 100 | 200 | 300 |
|---|---|---|---|
| *Training set* | | | |
| # Hidden Layers = 1 | 80.55% | 81.90% | 83.25% |
| # Hidden Layers = 2 | 88.40% | 91.26% | **92.45**% |
| # Hidden Layers = 3 | 87.33% | 90.65% | 92.18% |
| *Validation set* | | | |
| # Hidden Layers = 1 | 80.35% | 81.66% | 80.28% |
| # Hidden Layers = 2 | 86.20% | **90.75**% | 89.15% |
| # Hidden Layers = 3 | 86.33% | 89.57% | 88.72% |
| *Testing set* | | | |
| # Hidden Layers = 2 | – | **90.25**% | – |

Measures in bold show the best classification accuracy for different hyperparameter settings of hidden layers and numbers of neurons in the network. For this table, experimental results are reported using Twitter COVID-19 training, validation, and testing datasets

language model on three different datasets. For the Twitter dataset, we perform a set of experiments by changing the number of hidden layers and the number of hidden neurons as model parameters to achieve the best model selection. Detailed results are presented in Fig. 4 along with Tables 3 and 5.

## 5.1 Model Evaluation

Table 3 shows the classification accuracy measured using the k-fold cross-validation technique for our model selection study. The model selection procedure is described in Sect. 4.3. We conduct our experiment with different combinations of hyperparameters. For example, we raise the number of neurons from 100 to 300 and change the number of hidden layers from one to three.

The final results illustrate the training and validation set classification accuracy on different hidden layers and neuron settings as hyperparameters. With the validation set, we are able to determine a conclusion decision to nominate and select the best parameters for the dataset and present the final classification accuracy on the test set using selected parameters. We investigate and conclude that selecting the number of hidden layers as two and the number of neurons in the network as 200 generally produces the best classification performance on the validation set. Moreover, this performs well for testing set. For the remaining results reported in this paper, we use the same combination of parameters in our model.

We also observe that when the number of hidden layers increases beyond two, no such significant improvement is noticed in performance. Rather, this increases the training time and complexity of the model.

**Table 4** Statistical significance testing of algorithms for classification

| Algorithm | $p$ value |
|---|---|
| Proposed ensemble > Google | $1.37e{-}03$ |
| Proposed ensemble > Microsoft | $2.88e{-}05$ |
| Proposed ensemble > IBM | $4.15e{-}08$ |

$p$ values were calculated by pairwise binomial tests the on Twitter COVID-19 dataset. $C1$ ">" $C2$ indicates that $C1$ produces better results than $C2$ in a statistical manner

## 5.2 Statistical Significance Test

To investigate the statistical significance of the results given by each sentiment classifier, we run a binomial test [65] between pairs of every method. To understand the computation of the binomial test, let us assume that we have pairs of classifiers $C1$ and $C2$. Let $n$ be the number of records, where $C1$ and $C2$ provide different results. Let $s$ be the number of successes where classifier $C1$ predicts the correct sentiment label and $C2$ fails to do so, and $f$ be the number of times classifier $C2$ provides the correct sentiment label and $C1$ provides incorrect output. In this scenario, the $p$ value under binomial distribution can be written as

$$p\text{value} = \sum_{i=s}^{n} \frac{n!}{i!(n-i)!} \times p^i q^{n-i} \tag{7}$$

where $p$ and $q$ are the probabilities of success for classifiers $C1$ and $C2$, respectively. If we assume that there are no differences between methods, then $p = q = 0.5$ (null hypothesis). If the $p$ value is smaller than 0.05 (95% significance level), then we reject the null hypothesis and accept that classifier $C1$ is better than $C2$, as proved by statistics. Additionally, the smaller the $p$ value, the better the significance of a given result.

We notice that across all sentiment classifiers, our proposed ensemble model produces better classification results than individual models in a statistical manner. We report few $p$ value results using the Twitter COVID-19 dataset in Table 4.

## 5.3 Comparison to Other Methods

In Tables 5 and 6, we present the sentiment analysis results for our deep learning language model, Google sentiment analysis API, Microsoft sentiment analysis API, IBM sentiment analysis API, and the ensemble model across all the different datasets. We observe the same trend: our custom deep learning language model outperforms other existing models. Our model also shows an approximate 2% improvement over other models. Additionally, our ensemble deep learning language model (CustomDLL + Google + Microsoft

**Table 5** Comparative performance on sets 1 and 2 comprising Twitter, Amazon, and Yelp datasets

| Dataset | Custom DLL | Google | Microsoft | IBM | Proposed ensemble |
|---|---|---|---|---|---|
| Twitter | 90.25% | 87.10% | 88.25% | 84.40% | **92.65%** |
| Amazon reviews | 95.70% | 93.55% | 94.20% | 89.33% | **96.87%** |
| Yelp reviews | 96.66% | 95.28% | 95.90% | 94.90% | **97.50%** |

The results highlight our ensemble deep learning language model on sets 1 and 2 datasets. Our model consistently outperformed other existing classifiers

**Table 6** Comparative performance on set 3 comprising Web 2.0 datasets

| Dataset | Custom DLL | Google | Microsoft | IBM | Proposed ensemble |
|---|---|---|---|---|---|
| Twitter | 72.2% | 71.5% | 70.8% | 68.1% | **73.7%** |
| MySpace | 83.5% | 84.2% | 85.8% | 80.9% | **86.4%** |
| YouTube | 78.9% | 79.5% | 77.5% | 74.4% | **80.9%** |
| BBC | 31.4% | 29.7% | 30.5% | 27.1% | **35.8%** |
| Runners World | 76.6% | 78.2% | 77.4% | 71.5% | **80.8%** |
| Digg | 46.5% | 48.2% | 46.8% | 42.4% | **51.6%** |

The results highlight our ensemble deep learning language model on Web 2.0 dataset. Our model consistently outperformed other existing classifiers

+ IBM) shows better sentiment classification performance than others, with an improvement of approximately 2-5% in classification accuracy.

### 5.4 Runtime Performance

We measure the computational performance of our framework on the model training and prediction time. Our proposed custom DLL runs on a single workstation with an Intel-i7 1.8-GHz computer with a GPU and 32 GB of memory. The training time for our framework on first set coronavirus Twitter dataset was approximately 2.5 hours with pretrained weights. The prediction time of the proposed ensemble framework for one test record was approximately 10 seconds (because of the cumulative prediction of the Google, Microsoft, and IBM models). Similarly, the training time for second set for Yelp dataset was approximately 20 hours, and that for Amazon dataset was approximately 60. The average training time for third set that contains YouTube, MySpace, BBC, and others was 30 minutes due to small sample size.

## 6 Conclusions

In this paper, we proposed an ensemble deep learning language model that uses an advanced word embedding technique and creates an LSTM network for sentiment analysis. We evaluated our model on existing benchmarks with different settings of complexities and achieved better classification performance than the existing state-of-the-art sentiment analysis models. We assessed our model on a Twitter dataset specifically related to the coronavirus (COVID-19) pandemic to see how we can predict the sentiment of users by analyzing their tweets. The results suggested the potential of using our ensemble model for sentiment analysis. Further evaluation was performed on several social media application datasets, including Amazon, Yelp, YouTube, MySpace, BBC, and others.

We also indicated that our model produces decent classification accuracy when there are new words or terms present in the dataset, as in the case of coronavirus pandemic tweets. Our results verify that combining different individual classifiers and creating an ensemble classifier lead to improved classification performance. We performed a model selection experiment to investigate whether parameter settings were consistent across different datasets. In future work, we plan to update our model and incorporate several complementary features with the goal of improving the classification performance.

## References

1. Behera, R.K.; Naik, D.; Rath, S.K.; Dharavath, R.: Genetic algorithm-based community detection in large-scale social networks. Neural Comput. Appl. **32**(13), 9649–9665 (2020). https://doi.org/10.1007/s00521-019-04487-0.
2. Zhang, Z.; Sun, R.; Zhao, C.; Wang, J.; Chang, C.K.; Gupta, B.B.: CyVOD: a novel trinity multimedia social network scheme. Multimed. Tools Appl. **76**(18), 18513–18529 (2017). https://doi.org/10.1007/s11042-016-4162-z.
3. Gupta, S.; Gupta, B.B.: XSS-secure as a service for the platforms of online social network-based multimedia web applications in cloud. Multimed. Tools Appl. **77**(4), 4829–4861 (2018). https://doi.org/10.1007/s11042-016-3735-1.
4. Wang, H.; Li, Z.; Li, Y.; Gupta, B.B.; Choi, C.: Visual saliency guided complex image retrieval. Pattern Recognit. Lett. **130**, 64–72 (2020). https://doi.org/10.1016/j.patrec.2018.08.010.

5. Liu, B.: Text sentiment analysis based on CBOW model and deep learning in big data environment. J. Ambient Intell. Human Comput. **11**(2), 451–458 (2020). https://doi.org/10.1007/s12652-018-1095-6.
6. Pathak, A.R.; Pandey, M.; Rautaray, S.: Empirical evaluation of deep learning models for sentiment analysis. J. Stat. Manag. Syst. **22**(4), 741–752 (2019). https://doi.org/10.1080/09720510.2019.1609554.
7. Gupta, B.B.; Quamara, M.: An overview of Internet of Things (IoT): architectural aspects, challenges, and protocols. Concurr. Comput. Pract. Exper. (2020). https://doi.org/10.1002/cpe.4946.
8. Esposito, C.; Ficco, M.; Gupta, B.B.: Blockchain-based authentication and authorization for smart city applications. Inf. Process. Manag. (2021). https://doi.org/10.1016/j.ipm.2020.102468.
9. Alsayat, A.; Elmitwally, N.: A comprehensive study for Arabic sentiment analysis. Egyptian Inform. J. **21**(1), 7–12 (2020). https://doi.org/10.1016/j.eij.2019.06.001.
10. Al-Twairesh, N.; Al-Negheimish, H.: Surface and deep features ensemble for sentiment analysis of Arabic Tweets. IEEE Access **7**, 84122–84131 (2019). https://doi.org/10.1109/ACCESS.2019.2924314.
11. Dang, N.C.; Moreno-García, M.N.; De la Prieta, F.: Sentiment analysis based on deep learning: A comparative study. Electronics **9**(3), 483 (2020). https://doi.org/10.3390/electronics9030483.
12. Araque, O.; Corcuera-Platas, I.; Sánchez-Rada, J.F.; Iglesias, C.A.: Enhancing deep learning sentiment analysis with ensemble techniques in social applications. Expert Syst. Appl. **77**, 236–246 (2017). https://doi.org/10.1016/j.eswa.2017.02.002.
13. Carvalho, A.; Harris, L.: Off-the-Shelf technologies for sentiment analysis of social media data: two empirical studies. In: The Proceedings of 26th Americas Conference on Information Systems (AMCIS2020) **6**, Virtual Conference, Association for Information Systems (2020)
14. Zhang, L.; Wang, S.; Liu, B.: Deep learning for sentiment analysis: a survey. WIREs Data Min. Knowl. Discov. **8**(4), e1253 (2018). https://doi.org/10.1002/widm.1253.
15. Kamruzzaman, M.M.: Arabic sign language recognition and generating Arabic speech using convolutional neural network. Wireless Commun. Mob. Comput. **2020**, 3685614 (2020). https://doi.org/10.1155/2020/3685614.
16. Behdenna, S.; Barigou, F.; Belalem, G.: Document level sentiment analysis: a survey. EAI Endorsed Trans. Context-Aware Syst. Appl. (2018). https://doi.org/10.4108/eai.14-3-2018.154339.
17. Habimana, O.; Li, Y.; Li, R.; Gu, X.; Yu, G.: Sentiment analysis using deep learning approaches: an overview. Sci. China Inf. Sci. **63**(1), 111102 (2019). https://doi.org/10.1007/s11432-018-9941-6.
18. Whitehead, M.; Yaeger, L.: Sentiment mining using ensemble classification models. In: Sobh, T. (ed.) Innovations and Advances in Computer Sciences and Engineering, pp. 509–514. Springer, Dordrecht (2010)
19. Chen, N.; Wang, P.: Advanced combined LSTM-CNN model for Twitter sentiment analysis. In: The Proceedings of 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), IEEE, pp. 684–687 (2018) https://doi.org/10.1109/CCIS.2018.8691381
20. Behera, R.K.; Jena, M.; Rath, S.K.; Misra, S.: Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data. Inf. Process. Manag. **58**(1), 102435 (2021). https://doi.org/10.1016/j.ipm.2020.102435.
21. Stergiou, C.L.; Psannis, K.E.; Gupta, B.: B: IoT-based big data secure management in the fog over a 6G wireless network. IEEE Internet Things J. **8**(7), 5164–5171 (2021). https://doi.org/10.1109/JIOT.2020.3033131.
22. Heikal, M.; Torki, M.; El-Makky, N.: Sentiment analysis of Arabic tweets using deep learning. Proc. Comput. Sci. **142**, 114–122 (2018). https://doi.org/10.1016/j.procs.2018.10.466.
23. Cliche, M.: BB_twtr at SemEval-2017 task 4: twitter sentiment analysis with CNNs and LSTMs. In: The Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada, pp. 573–580 (2017) https://doi.org/10.18653/v1/S17-2094
24. Wang, X.; Jiang, W.; Luo, Z.: Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In: The Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, The COLING 2016 Organizing Committee, Osaka, Japan, pp. 2428–2437 (2016)
25. Armand, J.; Edouard, G.; Piotr, B; Tomas, M.: Bag of tricks for efficient text classification. arXiv preprint http://arxiv.org/abs/1607.01759
26. Luo, T.; Chen, S.; Xu, G.; Zhou, J.: Sentiment analysis. In: Trust-based Collective View Prediction, pp. 53–68. Springer, New York, NY (2013) https://doi.org/10.1007/978-1-4614-7202-5_4
27. Di Caro, L.; Grella, M.: Sentiment analysis via dependency parsing. Comput. Standards Interfaces **35**(5), 442–453 (2013). https://doi.org/10.1016/j.csi.2012.10.005.
28. Mouthami, K.; Devi, N. K.; Bhaskaran, M. V.: Sentiment analysis and classification based on textual reviews. In: The Proceedings of 2013 International Conference on Information Communication and Embedded Systems(ICICES), IEEE, pp. 271–276 (2013) https://doi.org/10.1109/ICICES.2013.6508366
29. Hu, H.; Phan, N.; Geller, J.; Iezzi, S.; Vo, H. T.; Dou, D.; Chun, S. A.: An ensemble deep learning model for drug abuse detection in sparse twitter-sphere. arXiv preprint https://arxiv.org/abs/1904.02062
30. Kumari, A.; Behera, R. K.; Sahoo, K. S.; Nayyar, A.; Kumar Luhach, A.; Prakash Sahoo, S.: Supervised link prediction using structured-based feature extraction in social network. Concurrency Computat Pract Exper, e5839(2020) https://doi.org/10.1002/cpe.5839
31. Alessia, D.; Ferri, F.; Grifoni, P.; Guzzo, T.: Approaches, tools and applications for sentiment analysis implementation. Int. J. Comput. Appl. **125**(3), 26–33 (2015). https://doi.org/10.5120/ijca2015905866.
32. Liu, B.; Zhang, L.: A survey of opinion mining and sentiment analysis. In: Aggarwal, C., Zhai, C. (eds.) Mining Text Data, pp. 415–463. Springer, Boston, MA (2012)
33. Yu, Y.; Duan, W.; Cao, Q.: The impact of social and conventional media on firm equity value: a sentiment analysis approach. Decision Support Syst. **55**(4), 919–926 (2013). https://doi.org/10.1016/j.dss.2012.12.028.
34. Kumar Behera, R.; Kumar Rath, S.; Misra, S.; Damaševičius, R.; Maskeliūnas, R.: Distributed centrality analysis of social network data using MapReduce. Algorithms (2019). https://doi.org/10.3390/a12080161.
35. Vohra, M.S.; Teraiya, J.: Applications and challenges for sentiment analysis: a survey. Int. J. Eng. Res. Technol. **2**(2), 1–6 (2013)
36. Das, S.; Behera, R.K.; Kumar, M.; Rath, S.K.: Real-time sentiment analysis of twitter streaming data. Proc. Comput. Sci. **132**, 956–964 (2018). https://doi.org/10.1016/j.procs.2018.05.111.
37. Wang, C.J.; Tsai, M.F.; Liu, T.; Chang, C.T.: Financial sentiment analysis for risk prediction. In: The Proceedings of the Sixth International Joint Conference on Natural Language Processing, Asian Federation of Natural Language Processing, Nagoya, Japan, pp. 802-808 (2013)
38. Fan, T.K.; Chang, C.H.: Blogger-centric contextual advertising. Expert Syst. Appl. **38**(3), 1777–1788 (2011). https://doi.org/10.1016/j.eswa.2010.07.105.

39. Yin, P.; Kamruzzaman, M.: Animal image retrieval algorithms based on deep neural network. Revista Científica de la Facultad de Ciencias Veterinarias **29**(2), 188–199 (2019)
40. Shukla, S.; Behera, R.K.; Misra, S.; Rath, S.K.: Software reliability assessment using machine learning technique. In: Chakraverty, S., Goel, A., Misra, S. (eds.) Towards Extensible and Adaptable Methods in Computing, pp. 57–68. Springer, Singapore (2018)
41. Chen, X.; Zhang, L.; Liu, T.; Kamruzzaman, M.: Research on deep learning in the field of mechanical equipment fault diagnosis image quality. J. Vis. Commun. Image Represent. **62**, 402–409 (2019). https://doi.org/10.1016/j.jvcir.2019.06.007.
42. Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D.: Text classification algorithms: a survey. Information **10**(4), 150 (2019). https://doi.org/10.3390/info10040150.
43. Ma, A.; Liu, Y.; Xu, X.; Dong, T.: A deep-learning based citation count prediction model with paper metadata semantic features. Scientometrics **126**(8), 6803–6823 (2021). https://doi.org/10.1007/s11192-021-04033-7.
44. Mittal, V.; Gangodkar, D.; Pant, B.: Deep graph-long short-term memory: a deep learning based approach for text classification. Wireless Pers. Commun. **119**(3), 2287–2301 (2021). https://doi.org/10.1007/s11277-021-08331-4.
45. Deepika, N.; Nirupama Bhat, M.: An efficient stock market Pprediction method based on kalman filter. J. Inst. Eng. India Ser. B **102**(4), 629–644 (2021). https://doi.org/10.1007/s40031-021-00583-9.
46. Hasni, S.; Faiz, S.: Word embeddings and deep learning for location prediction: tracking coronavirus from British and American tweets. Soc. Netw. Anal. Min. **11**(1), 66 (2021). https://doi.org/10.1007/s13278-021-00777-5.
47. Johnson, R.; Zhang, T.: Effective use of word order for text categorization with convolutional neural networks. In: The Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, pp. 103–112 (2015) https://doi.org/10.3115/v1/N15-1011
48. Chen, G.; Wang, L.; Kamruzzaman, M.: Spectral classification of ecological spatial polarization SAR image based on target decomposition algorithm and machine learning. Neural Comput. Appl. **32**(10), 5449–5460 (2020). https://doi.org/10.1007/s00521-019-04624-9.
49. Akhtar, M. S.; Kumar, A.; Ghosal, D.; Ekbal, A.; Bhattacharyya, P.: A multilayer perceptron based ensemble technique for fine-grained financial sentiment analysis. In: The Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, pp. 540–546 (2017) https://doi.org/10.18653/v1/D17-1057
50. Yanmei, L.; Yuda, C.: Research on Chinese micro-blog sentiment analysis based on deep learning. In: The Proceedings of 2015 8th International Symposium on Computational Intelligence and Design (ISCID) **1**, IEEE, pp. 358–361 (2015) https://doi.org/10.1109/ISCID.2015.217
51. Rida-E-Fatima, S.; Javed, A.; Banjar, A.; Irtaza, A.; Dawood, H.; Dawood, H.; Alamri, A.: A multi-layer dual attention deep learning model with refined word embeddings for aspect-based sentiment analysis. IEEE Access **7**, 114795–114807 (2019). https://doi.org/10.1109/ACCESS.2019.2927281.
52. Rehman, A.U.; Malik, A.K.; Raza, B.; Ali, W.: A Hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis. Multimed. Tools Appl. **78**(18), 26597–26613 (2019). https://doi.org/10.1007/s11042-019-07788-7.
53. Al-Makhadmeh, Z.; Tolba, A.: Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. Computing **102**(2), 501–522 (2020). https://doi.org/10.1007/s00607-019-00745-0.
54. Wen, L.; Hughes, M.: Coastal wetland mapping using ensemble learning algorithms: a comparative study of bagging, boosting and stacking techniques. Remote Sens. (2020). https://doi.org/10.3390/rs12101683.
55. Altman, N.; Krzywinski, M.: Ensemble methods: bagging and random forests. Nat. Methods **14**(10), 933–935 (2017)
56. Zaman, M.F.; Hirose, H.: Classification performance of bagging and boosting type ensemble methods with small training sets. New Gener. Comput. **29**(3), 277 (2011). https://doi.org/10.1007/s00354-011-0303-0.
57. Ardabili, S.; Mosavi, A.; Várkonyi-Kóczy, A.R.: Advances in machine learning modeling reviewing hybrid and ensemble methods. In: Várkonyi-Kóczy, A. (ed.) Engineering for Sustainable Future. Lecture Notes in Networks and Systems, pp. 215–227. Springer, Cham (2020)
58. Sherstinsky, A.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D Nonlinear Phenomena **404**, 132306 (2020). https://doi.org/10.1016/j.physd.2019.132306.
59. Palagin, O.; Velychko, V.; Malakhov, K.; Shchurov, O.: Distributional semantic modeling: a revised technique to train term/word vector space models applying the ontology-related approach. arXiv preprint https://arxiv.org/abs/2003.03350v1
60. Yu, Y.; Si, X.; Hu, C.; Zhang, J.: A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. **31**(7), 1235–1270 (2019). https://doi.org/10.1162/neco_a_01199.
61. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint https://arxiv.org/abs/1810.04805
62. Ferrucci, D.; Levas, A.; Bagchi, S.; Gondek, D.; Mueller, T.E.: Watson: Beyond Jeopardy! Artif. Intell. **199–200**, 93–105 (2013). https://doi.org/10.1016/j.artint.2012.06.009.
63. Al-Makhadmeh, Z.; Tolba, A.: Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. Computing **102**(2), 501–522 (2020). https://doi.org/10.1007/s00607-019-00745-0.
64. Thelwall, M.: The heart and soul of the web? sentiment strength detection in the social web with sentistrength. In: Holyst, J. (ed.) Cyberemotions. Understanding Complex Systems, pp. 119–134. Springer, Cham (2017)
65. Salzberg, S.L.: On comparing classifiers: pitfalls to avoid and a recommended approach. Data Min. Knowl. Discov. **1**(3), 317–328 (1997). https://doi.org/10.1023/A:1009752403260.