

# RDNN: Rumor Detection Neural Network for Veracity Analysis in Social Media Text

**SuthanthiraDevi P<sup>1\*</sup>, and Karthika S<sup>2</sup>**

<sup>1</sup>Department of Information Technology, St Joseph College of Engineering,  
Kalavakkam – 603110, Tamil Nadu, India  
[e-mail: devisaran2004@gmail.com]

<sup>2</sup>Department of Information Technology, SSN College of Engineering,  
Kalavakkam – 603110, Tamil Nadu, India  
[e-mail: skarthika@ssn.edu.in]

\*Corresponding author: SuthanthiraDevi P

*Received June 11, 2022; revised October 25, 2022; accepted November 15, 2022;  
published December 31, 2022*

---

## Abstract

A widely used social networking service like Twitter has the ability to disseminate information to large groups of people even during a pandemic. At the same time, it is a convenient medium to share irrelevant and unverified information online and poses a potential threat to society. In this research, conventional machine learning algorithms are analyzed to classify the data as either non-rumor data or rumor data. Machine learning techniques have limited tuning capability and make decisions based on their learning. To tackle this problem the authors propose a deep learning-based Rumor Detection Neural Network model to predict the rumor tweet in real-world events. This model comprises three layers, AttCNN layer is used to extract local and position invariant features from the data, AttBi-LSTM layer to extract important semantic or contextual information and HPOOL to combine the down sampling patches of the input feature maps from the average and maximum pooling layers. A dataset from Kaggle and ground dataset #gaja are used to train the proposed Rumor Detection Neural Network to determine the veracity of the rumor. The experimental results of the RDNN Classifier demonstrate an accuracy of 93.24% and 95.41% in identifying rumor tweets in real-time events.

---

**Keywords:** Attention mechanism, Bidirectional Long Short Term Memory (Bi-LSTM), Convolution Neural Network (CNN), Deep Learning, Natural Language Processing.

## 1. Introduction

**S**ocial media platforms such as Facebook and Twitter generate various types of content. Twitter is one of the most popular microblogging services to share information online [4]. The information posted on Twitter will reach millions of people within a second [1]. A sizable quantity of information shared on Twitter is a combination of textual and visual content, namely images and videos. Veracity is to determine whether the information posted on social media is credible or not. Misinformation on social media can instigate panic among people during crises. The velocity of dissemination of credible content is much slower than that of counterfeit content. To reduce the causes of the rumor can be reduced by applying deep learning models.

Ahmad et.al [2] introduced a terrorist-based investigation structure to point out organizing extremist and non-extremist classes using deep learning models like CNN-LSTM which achieves greater accuracy than baseline models. Conventional feature sets of the content are filtered with the aid of bag-of-words (BoW), Tf-IDF and word embedding. The learned features are embedded using CNN- LSTM, GRU and Fast Text. However this model failed to take visual and social context features into consideration and lacked an automated method to store Twitter content. Oluwaseun et al., [17] introduced an outline for identifying rumor messages from Twitter using the integration of CNN and RNN model. This model finds related features associated with rumor reports without referring to the prior content. The author concluded that to increase the size of the training dataset and thus improve the model's robustness. Alzanin et al., [3] surveyed how rumors in micro-blogging websites had been done through three divisions namely supervised, unsupervised and hybrid approaches in machine learning-based models. For supervised learning each tweet is labeled by a group of human annotators based on their ground truth and chats. The level of the fake news is determined by building a classifier with a set of features. The supervised classifier of J48 decision tree has achieved a good result of 86% of accuracy. Unsupervised approaches utilizing the clustering analysis and the rumors are identified based on the characteristics like retweet ratio and verified user. This classifier fails to investigate rumors with multilingual features.

Hybrid approaches combine the features of the clustering and classification models. Chang et al., [6] focused on non-rumor information, reliability and also introduced a rule-based model for distinguishing political rumors from Twitter for managing political campaigns during public elections depends on recognizing intense users. They have focused on the clustering method to identify fake tweets from Twitter. Extreme users(to spread rumors constantly on Twitter) are identified by combining the structural and timeline features. A rule-based model was ineffective to generate topic and extreme user keywords automatically. Chen et al., [7] invented an attention-based deep learning method termed a recurrent neural network for classifying rumors via obtaining knowledge of hidden information from tweet posts automatically. CallAtRumors method evaluating metrics like precision, recall, and F1-score and cross-topic emerging techniques for classifying Twitter datasets. However, the Deep Attention model does not deal with the complex features of the rumors. Guo et al., [10] introduced Hierarchical Bi-directional Long Short Term Memory (Bi-LSTM) to detect rumors at different levels (early detection or post-detection) from social context. An attention mechanism was developed for assimilating social context to the network in addition to that value for early recognition of rumors as well as rumor identification. These models

investigate only small quantities of the posts. Gao et.al. [29] presented an early rumor detection model using a hybrid deep learning architecture. This model combines the bidirectional language model with stacked LSTM networks. It can analyze the tweet content and propagation context to identify rumor diffusion at early stages. Out-of-date ratings may become noise details for recommendations. Deng et al., [27] explores the Deep Learning based Matrix Factorization (DLMF) method to identify the trust based community in social networks. The objective function learnt the features of the trust community and user characteristics in the network. The author does not investigate the efficiency and scalability of online social networks. Han et al., [11] utilized two stages in finding rumors and information published in tweets by users. The first stage exposed CNN and in the second phase, Bi-LSTM was used for constructing memory units for categorizing the tweets. However this system captures local information and sequence information. Zhao et al., [25] determines the integration of attention-based CNN can confine contextual information for every word posted in tweets. Indeed attention between word pairs need to incorporate rich information into the attention mechanism. The main objective of this research work is to detect rumors in Twitter data during natural disaster times. This task faces a variety of challenges, including various types of rumors, various events, and multiple ways of representing the same semantics. Existing rumor informatics works are based on classical machine learning techniques or use classical feature representation schemes followed by a classifier. The research methodology detects rumors in real-world events such as elections, sporting events, earthquakes, floods, and blasts. The spread of untrustworthy content online has resulted in financial and infrastructure losses, as well as a threat to human lives in the offline world. The following are the contributions made in this work

- Categorizing tweets as rumors or non-rumors association with deep learning-based Rumor Detection Neural Network technique.
- Scrutinizing conventional vectorization methods for TF-IDF, CV for machine learning algorithms, and pre-trained vectorization methods such as Glove, Fast Text for neural network tweet categorization of rumor data and non-rumor data.
- The proposed RDNN model outperforms existing methods in terms of enhanced accuracy.
- To exhibit the effectiveness of the model, the proposed model is compared with existing methods.

For rumor news detection many techniques have been implemented and are discussed in section 2. Machine learning and deep learning algorithms with proposed algorithms are described in section 3. Section 4 covers the corpus collection, data annotation, and pre-processing steps. The machine learning algorithm and proposed model results for the detection of rumor messages are discussed in section 5. The performance analyses of the proposed system with the existing systems are discussed in section 6. Section 7 concludes the proposed model along with future work.

## 2. Related Work

Recently, many deep learning techniques have been developed for detecting rumors in social media. In this section, the techniques related to deep learning models for detecting rumors on social media are discussed.

Zeng et al., [25] offered the LSTM-CNN model for language-specific issues in trigger labeling. In this survey, Zeng proposed novel techniques to extract Chinese events without using any natural language processing tools such as POS, NER. The Chinese trigger labeling

was aimed to discover predefined event types. Error propagation problem was reduced by integrating trigger identification and type classification into one neural network. Chinese words do not have a delimiter between words. These kinds of issues were eradicated by using cross-world triggers inside word triggers. However, this approach is not capable of identifying some of the arguments which are triggered in similar sentences. Veyseh et al., [20] investigated a new approach CNN-LSTM to identify the position of tweets against humorous tweets. It incorporates external relational features such as friendship in a thread of conversation from tweets. This model achieves higher accuracy with an F1-score of 3.6 for the SemEval dataset. Liu et al., [13] established attention-based Bi-GRU with CNN and RNN were used to explain issues in classifying Chinese university question papers. Prasannata patwa et al., [18] identified rumors on social media that have caused damage to several companies in India. Predicament at Infibeams, Fab hotels, and Kalyan jewelers have lost their profit because of rumors messages spread over social media like Facebook, Twitter, etc. For instance, Infi-beam Avenues Ltd, an online marketplace lost its market value 71% in a single day on account of a rumor message. CNNs outperform SVMs and fully connected neural networks on these datasets.

Cornelia et al., [6] used CNNs for text classification in the task of identifying informative tweets during natural disasters. The automated detection of informative data within micro blogging platforms were investigated. The authors demonstrate rumor messages empirically on several real-world flooding cases. Mohammed et al., [12] proposed a novel transformation mechanism named as Binary Neural Network (BNet). This is a data-driven, end-to-end neural-based model that does not rely on external resources like sentiment, parts of speech taggers and emotion lexicons.

Manzhu et al., [24] compares the performance of the CNN model for cross-event topic classification with two traditional machine learning models, SVM and LR, using three geo-tagged Twitter datasets collected during Hurricane Irma, Hurricane Sandy and Harvey. Specifically, two groups of experiments are carried out a single-event experiment in which 80% of the dataset from each disaster event is randomly selected to build the classifiers and the remaining 20% is used to test the performance of each classifier, A cross-event validation experiment in which classifiers trained with historical Twitter datasets are compared to classify tweet messages generated during a later event into different topics. CNN was one of the most widely used pattern recognition algorithms. The experimental results show that the proposed CNN-based model can build an evolving SA knowledge base using historical Twitter datasets to help with topic classification for future hurricane events with insufficient training data.

Kim [12] introduced CNN for sentence-level classification tasks using pre-trained word vectors. Convolving a text matrix with detection filters of varying lengths was used to search for the presence of specific features or patterns in the text. Then, using max pooling and the extractive vectors of each filter, the input representation was down-sampled by reducing its dimensionality and allowing assumptions to be made about features contained in the filtered sub regions. As a result, each filter corresponds to a digit, and these filters form a vector representing a sentence. Zhang [34] used temporal convolutional networks to apply deep learning to text understanding and ran experiments on various large-scale datasets, including ontology classification, sentiment analysis, and text categorization. The results demonstrated that temporal ConvNets can perform satisfactorily without knowledge of words, phrases, sentences, or any other syntactic or semantic structures in a human language. ConvNets were only applied to characters in large-scale datasets, and the results showed that deep ConvNets can classify words without knowledge of words or the syntactic or semantic structure of a

language. The taxonomy of deep learning-based methods for veracity analysis in Twitter is explained in the following [Table 1](#).

**Table 1.** Taxonomy of deep learning-based algorithms for veracity analysis in Twitter

Authors	Dataset	Techniques used	Deep Learning layers	Proposed Approach
Caragea et al [5]	CrisisLexT26 Project dataset	Convolutional Neural Networks	Convolutional Layer, Pooling Layer, Output layer	CNN approach is presented to identify informative messages in social networks during crisis events.
Graves et al [8]	Penn Treebank portion of the Wall Street Journal corpus, Hutter Prize Wikipedia datasets	Recurrent Neural Networks	Deep RNN composed of stacked LSTM layers	Developed an RNN-based model to generate real-valued and discrete sequences with long-range structure.
et al [11]	SemEval-2018 Task1 dataset	Deep Neural Network	Embedding layer, RNN +GRU, Dropout layer	Deep learning model to classify multiple emotions in Twitter.
Kim et al [12]	Movie reviews, Stanford Sentiment Treebank, TREC question dataset, Customer reviews of various products, MPQA dataset	Convolutional Neural Networks	Convolutional layer, Max-over-time Pooling, Fully connected layer	Classification of sentences with the aid of pre-trained word vectors.
Severyn et al [19]	Twitter 13 dataset, Twitter15 dataset	Deep Neural Network	Sentence matrix with activations, Convolutional, Pooling, and Softmax layers.	DNN for Twitter Sentiment Analysis
Yubo et al [21]	ACE 2005 corpus	Dynamic Multi-pooling Convolutional Neural Network(DMCNN)	Convolution layer, Dynamic Multi-Pooling	A DMCNN is developed to extract events from Twitter sentences.
Zhang et al [23]	DBPedia ontology dataset, AG's news corpus, Sogou news corpus,	Convolution Neural Network	6 Convolutional layers and 3 Fully-connected layers.	A Twitter text was classified using character-level convolutional networks.
Proposed Method	#gaja cyclone Tweet, Kaggle Dataset	Attention mechanism, CNN-Bi-LSTM with Hybrid Pooling	Attention Layer, Convolutional Layer, Bi-LSTM, Hybrid-pooling Layer, Dense layers	RDNN model is developed to detect rumor tweets using crisis event tweet messages

### 3. Dataset Description

#### 3.1 Regarding Datasets

**Dataset 1:** The tweets related data are accumulated from Cyclonic Storm Gaja under various hash tags and user IDs namely, #cyclonegaja, @TNSDMA, and #SaveDelta. The tweets related information are collected over a two weeks period from 12<sup>th</sup> Nov to 25<sup>th</sup> Nov during the year 2018 using tweepy.api. Twitter data is textually unstructured which contains images, URLs, and video files. The inconsistent twitter data has been analyzed and tweets data are

manually annotated. Finally statistical analysis of #gaja [33] had been done for three hashtags that are discussed in Table 2. It is a total number of counts measured in parameters. An integral part of the data is analyzed by the unit of the total count. The statistics of the tweet data are mentioned as counts of total number of tweets, unique users, images, videos and URL information about the tweet. A ground dataset called "#gaja" is generated. The tweets were collected from the Cyclonic Storm Gaja crisis event using various hashtags and user IDs, including #cyclonegaja, @TNSDMA, and #SaveDelta. The tweet data was cleaned using preprocessing rules. With the help of data annotation, the tweets are labeled as rumor or non-rumor. The tweets have two levels of annotation: primary and secondary. First, tweets about the event are filtered, and then they are classified as rumor or not based on their credibility [26]. The authors use the Fleiss kappa coefficient to validate their data annotation work. It is used to calculate the Inter Rated Reliability which uses three annotators to classify tweets [35].

**Dataset 2:** The fake news datasets in Twitter gathered from Kaggle [32] resources and perform statistical analysis of distinguishing rumor and non-rumor data and details are shown in Table 3. The images and video content are removed in order to identify the rumor messages,

**Table 2.** Statistics for #gaja tweet data corpus

Total Number of tweets	Count of Total unique users	Count of Tweets with URL	Count of Tweets with fake images	Count of Tweets with real images	Count of Tweets with fake videos	Count of Tweets with real videos	Count of Tweets with unknown URLs
60,572	26,573	14,916	281	4929	64	2340	4566

**Table 3.** Statistical features analysis usage from Gaja & Kaggle

Twitter Dataset	Count of Total number of tweets	Count of Number of rumor tweets	Count of Number of non-rumor tweets
#gaja	90897	63606	27260
Kaggle	6315	3161	3154

### 3.2 Data Annotation

The tweet data from #gaja incidents were categorized into three classes namely (1) Definitely Real data (DR) (2) Partially Real data (PR) and (3) Definitely Fake data (DF). The description of each class tweets with examples are illustrated in Table 4. The DR and PR data are considered as Rumor and DF as Non-rumor data.

**Table 4.** A sample tweet for each type of rumor

Class	Description	Example Tweet
Definitely Real data (DR)	A news from official government source IMD	Gale wind speed reaching 70-80 kmph gusting to 85 kmph prevails over East central and adjoining West central & Southeast Bay of Bengal.



Partially Real data (PR)	Information about a related incident	#GAJA is very likely to further strengthen during the next few days and become a severe cyclonic storm over the Bay of Bengal.
Definitely Fake data (DF)	Unverified information	Schools and colleges in Tamil Nadu remained shut today and offices and business establishments were asked to relined?

### 3.3. Preprocessing

The dataset is collected from the twitter using #gaja hashtag during crisis period. This data contains URLs, emoticons and smiley symbols. It also includes a lot of noises in the form of unstructured data like images and abbreviations. Reducing the noise in the text helps to improve the performance and speed up the classification process. The Natural Language Processing technique used to implement the preprocessing rules. The authors have applied the nine preprocessing rules to preserve the meaning of the data [28]. The rules are, convert to lowercase, RT removal, Replacement of User-mentions, URL replacement, Hash character removal, Removal of punctuations and symbols, Lemmatization, Stopword removal and Unwanted whitespaces removal. **Table 5** illustrates original tweet from #gaja dataset along with pre-processed tweets. The original tweet contains the special characters @ and #, punctuations and symbols, RT, uppercase letters and Unwanted whitespaces. After preprocessing RT, @, # and Unwanted whitespaces are removed. The user name of tweet '@ndmaindia' is replaced as user. The preprocessed data to improve the RDNN model performance.

**Table 5.** Pre-processed Tweet example

Original Tweet	RT @ndmaindia: #CycloneGaja #Wind Warning: Gale wind speed reaching 70-80 kmph gusting to 85 kmph prevails over East central and adjoining W?
Pre-processed tweet	user cyclone gaja wind warning gale wind speed reaching kmph gusting to kmph prevails over east central and adjoining W

For deep learning models, the tweet texts are preprocessed using tokenized functionality. Each sentence has to be split into words in order to train the neural networks. The words are converted into a sequence of integer values. It can filter the most common word based on a list of texts available in internal vocabulary. Only the 'N' number of most common words applied to neural networks leads to an increase in execution time. The number of frequencies of each word is calculated and we select the most common words from the dataset. Padding generates sequences of integers which contain similar size for both training data and testing data.

### 3.4 Feature Extraction

The transformation of unstructured textual data into structured data is referred to as feature extraction. The feature extraction phase is used to extract the meaningful tokens from the tweet messages. The authors have applied the techniques of Term Frequency-Inverse Term

Frequency (TF-IDF), Count Vectorizer (CV) and Word Embedding to extract the features. The representation of TF-IDF is expressed using formula (14)

$$Tf - idf = tf(t, d) * idf(t, d) \quad (1)$$

Where 't' represents the number of tokens in the document d. Countvectorizer is used to count the most frequently used tokens in the data. These techniques are applied to the classifiers of Naïve Bayes, Support Vector Machine, Logistic Regression, and Decision tree. The instigator employed a word embedding method for transforming the unstructured data into a feature vector through training datas by neural networks. A word vector representation s shown as equation (15) as

$$[w, y1, y2, \dots, yn] \quad (2)$$

where 'w' represents word and y1, y2... symbolize dimensions of w indicating numerical value. This vector is generated based on semantic similarity between word features.

For instance

$$\begin{aligned} \{\text{Actor, Actress}\} &= \{\text{Man, Woman}\} \\ \{\text{France, India}\} &= \{\text{Paris, Delhi}\} \\ \{\text{Red, Orange}\} &= \{\text{Oil, Milk}\} \end{aligned}$$

DNN categorizes the concepts similar to context that are arranged into one embedded space and other not. It processes the actual sequence of words. The order and context of words are maintained as it appears in the document. The word sequences are converted into the word index vector. For e.g. the word index of words are calculated as follows

[Schools : 1 ,and : 2, colleges : 3, in : 4 ,Tamil Nadu : 5, remained : 6 , shut : 7, today : 8]

**Table 6.** Word Index Vector

Word Sequence	Word Index vector
[Schools and colleges in Tamil Nadu remained shut today]	[1, 2, 3, 4, 5, 6, 7, 8]
[Today Schools and colleges in Tamilnadu shut]	[8, 1, 2, 3, 4, 5, 7]

**Table 6** depicts the #gaja dataset word sequence with respect to the word index vector. These sequences can be directly fed into a deep neural network for training and classification.

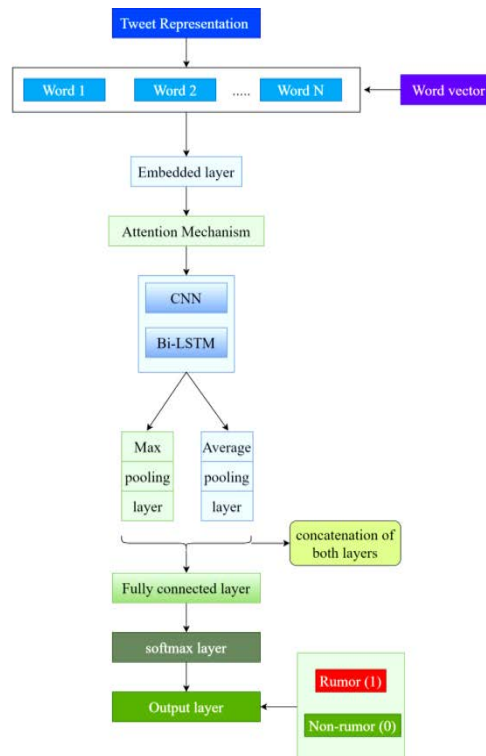
## 4. Methodology

The methodology explains how the proposed work is implemented with the attention mechanism for classifying social textual events into rumors and non-rumors. The rumor tweets are to be predicted by the proposed RDNN model with nine layers of the Deep-neural network. A RDNN model can be created using nine layers namely embedding layer, attention layer, convolution layer, Bi-LSTM layer, concatenated max-pooling and average pooling layer and fully connected layer and dense layer with two activation functions. The overall architecture of event-tagged rumor identification is depicted in the following **Fig. 1**. First layer is the embedding layer which converts the preprocessed text as a dense vector of continuous values. The Attention Convolution layer can extract features from the embedding layer and then pass into hybrid\_max\_pooling layer. Two dense layers with activation functions are used to categorize rumors happening in that occurrence #gaja.



#### 4.1 Deep Learning model for Rumor Detection Neural Network (RDNN)

The proposed model RDNN architectural diagram for detecting rumors from twitter is depicted in **Fig. 1**. This model is implemented using the packages Python, Keras and Tensorflow. An attention mechanism is used to get the context information from the tweet messages. In order to capture salient local lexical features, CNN is used without using Parts of Speech Tagging (POS tags) (or) Named Entity Recognition (NER). conv1D is a mathematical operation in python language which is used to summarize a feature vector into a one dimensional array. Bi-LSTM is helpful in encoding the semantic meaning of words in whole tweet sentences. Convolutional Neural Network is mainly used to filter local features between words embedding vector size 512. The filter sizes are 4, 5 with 6 feature maps. Each stride value is set to 1 and padding value is set to 0. The pooling layer is critical in extracting the most important features and discarding some useless and irrelevant features from each twitter's sentences. In this experiment, the authors have investigated by integrating both average pooling and maximum pooling layers to capture missing richer feature information in the entire dataset. Combined features are applied to fully connected layers. Finally, the softmax classifier is used to distinguish rumor from non-rumor (abnormal) i.e. rumor data from the sentence. **Fig. 1**, describes the proposed RDNN algorithm and their implementation details.



**Fig. 1.** Architecture for rumor detection using Gaja dataset & twitter dataset from Kaggle

#### 4.2 Attention Mechanism

The generated feature vectors are fed into the attention layers. An attention mechanism is an

enhanced version of encoder-decoder based on neural network systems in NLP processing. The encoding mechanism for the attention mechanism is shown in Fig. 2. First attention model was proposed by Bahdanau et al., [4] for performing neural machine translation. It has two components of encoder-decoder integrated with Recurrent Neural Network and LSTM which helps to generate better performance. Encoder LSTM admits a particular component of input sequence as a collection of words converted into semantic vectors. Each word is represented as  $x_i$ , where 'i' refers to the order of the word.

Basically in Convolutional Neural Network, the hidden states of  $h_i$  are computed by using equation

$$h_t = f(W^{hh}h_{t-1} + W^{hx}x_t) \tag{3}$$

This formula stands for outcome of an ordinary recurrent neural network. Here  $x_i$  is input vector  $h_{t-1}$  refers to the hidden state. The encoder vector is the final hidden state generated from the encoder part as well as initial hidden state for the decoder portion which makes precise prediction of the model. The hidden state that occurs in CNN has been calculated using formula 6. The decoder decodes all information gathered from the encoder vector and generates output sequences consisting of a collection of words. Any hidden state in CNN is calculated using formula

$$h_t = f(W^{hh}h_{t-1}) \tag{4}$$

The output at time 't' is determined using the formula

$$y_t = softmax(W^sh_t) \tag{5}$$

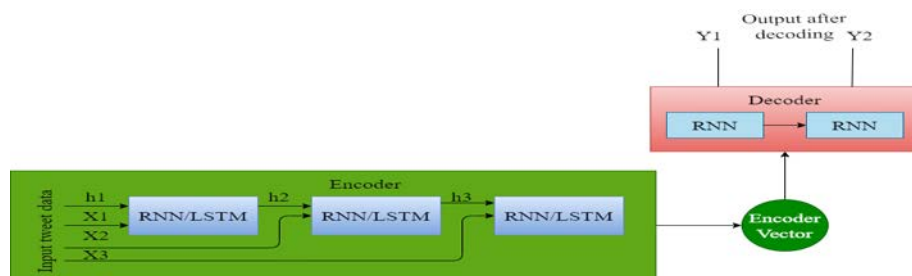


Fig. 2. Attention Mechanism to encode tweet data, decode using encoder vector

The equations are utilized [14] for explaining the concept of attention mechanism. Liu et al., [14] has used the attention mechanism for sentiment analysis and achieved better accuracy values for sentiment classification. Each sentence has a different degree of weights and attention mechanism to extract semantic features from the sentence. The authors make use of attention to extract the semantic information from the dataset. The attention mechanism is evaluated using the attention formula by equation [8, 9].

$$C_t = \sum_{i=1}^{T_h} a_{ij} \tag{6}$$

$$a_{ij} = softmax(W_{a2} tanh(W_{a1} H_t)) \tag{7}$$

Where  $a_{ij}$  represents size of attention weight,  $T_h$  denotes sequence data length, and weights are represented as  $W_{a1}$ ,  $W_{a2}$ .

### 4.3 Convolution Layer

This layer learns patterns from text sequences generated by the embedding layer. It has extracted useful features in an intense structure and deemed only most relevant features for rumor classification. The architecture of the CNN is depicted in Fig. 3. A convolution neural

network derives the features of all overall dataset into fixed-length segment words that are generated for training. The immense benefit of using this layer is dimensionality reduction of the dataset and that it consumes less execution time. Guo et al., [9] A new feature is generated by using the automatically learned weights during the training. The weight matrix 'w' is applied to a sequence of 'k' words and new feature  $a_i$  is generated by

$$a_i = f(w * x_{i:i+k-1} + b) \tag{8}$$

where 'b' is constant which is referred to as bias and f is a nonlinear function. Equation (10) is applied to each window of words in the text array of  $\{x_{1:n}, x_{2:n+1}, \dots, x_{n-k+1:n}\}$  then the feature map is derived as

$$A = [a_1, a_2, \dots, a_{n-k+1}] \tag{9}$$

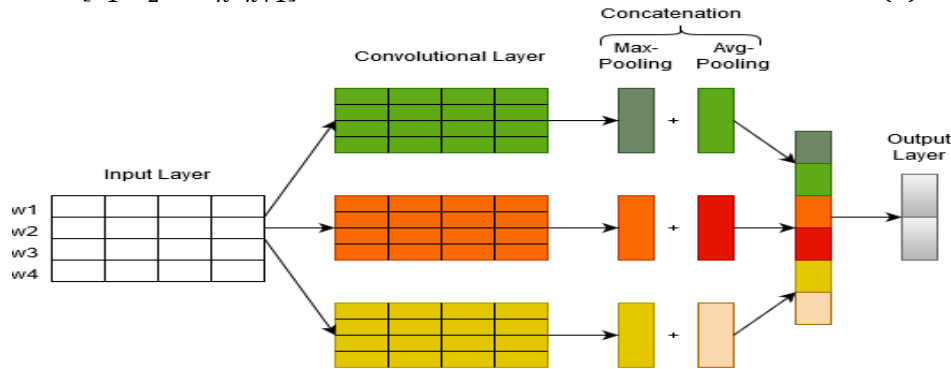


Fig. 3. Architectural diagram for Convolutional Neural Network

The ReLU activation function performs non-linear transformation to input data that enables learning and execution operation. It can decide whether the information received from input data is pertinent or not. In this operation, the convolution layer accepts word embedding vector size of  $5000 \times 300$ . It uses a batch size of 128, length of word as 5 for each sentence. The CONV\_1D network is evaluated with the ReLU activation function. The output size of the embedding vector is reduced in to  $4996 \times 128$  matrix size.

#### 4.4 Bi-LSTM

The Bi-LSTM layer to extract the semantic features from the sentence vector. This model summarizes the whole tweet data by deriving the hidden state of each word in both backward and forward direction as shown on Fig. 4.

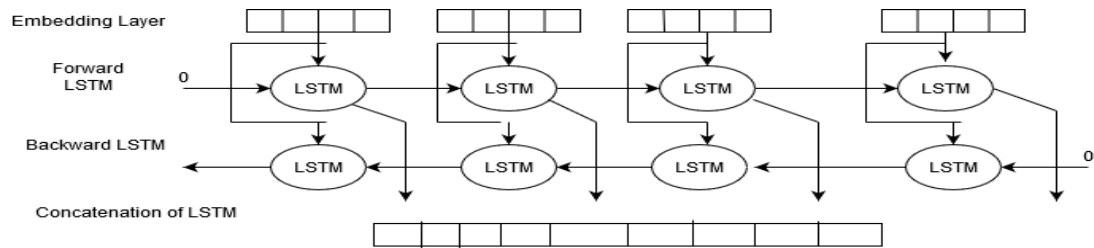


Fig. 4. Bi-LSTM model Recurrent Neural Network for embedding forward and backward layer

The input sentences are executed in two ways as in the form of previous to next word and next to previous word prediction. This helps to preserve information in order to retrieve it at any time in future need. The output of the layer is fed into the pooling layer.

#### 4.5 Hybrid max\_pooling layers

In RDNN model, pooling layers are added to extract the important features of the rumor data. This layer extracted the most important features from the feature map [13,25].

$$P_j = \text{concatenate} \left( \max_{\text{pooling}} (A, \text{padding}), \text{avg}_{\text{pooling}} (A, \text{padding}) \right) \quad (10)$$

The main advantage of using this layer is to down sample i.e entire feature map into a single value. It can reduce input dimensionality via dimensionality reduction and decrease the number of computations or parameters which is for further usage in social networks. In our experiment the global max pooling has produced a single value of 128 and it can summarize all features into a single feature map.

#### 4.6 Dense layers

A feature map generated by the pooling layer is connected to the dense layers. This layer is supportive to solve linear operations occurring in the network. In this layer every input is connected to every other output with some weights which is represented by matrix vector multiplication. During back propagation, the matrix values with trainable parameters are updated and it produces 'm' dimensional vectors which can have capability to modify the dimension of the input vector. In our investigation the output of this layer reduces the parameters into 16512 and 129. The reduced features are applied to the activation layer.

#### 4.7 Activation layers

Activation layers determine which layers of the mode will be activated. The most significant feature of artificial neural networks is active function. This function decides network activity whether a neuron should be activated or not. It can determine the output of the model, its accuracy and computational efficiency of the training model. Activation helps to normalize the output of each neuron ranging from one to zero. A non-linear Rectified Activation function (ReLU) was introduced by Nair et al., [16] The ReLU function is derived using equation (11) as follows

$$F(X) = \max(0, x) \quad (11)$$

The ReLU function has bounded values only for negative input data since this function activates only a few neurons at certain times making computation very proficient. The softmax activation function has normalized output values between 0 and 1. Also, the main aspiration of softmax function is to filter out large values and suppress below large values which makes it easier to handle multiclass classification. The softmax function is estimated using equation (13)

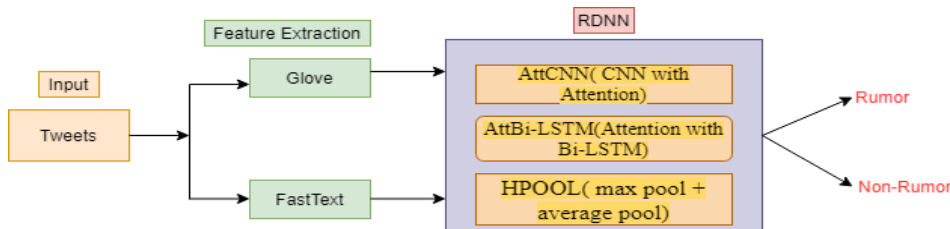
$$\text{Softmax}(X)_j = \frac{e^x}{\sum_{n=1}^N n e^x} \quad (12)$$

The classifier model is compiled with the aid of a cross-entropy loss function. This function has summarized the average distribution between actual values as well as predicted values.

#### 4.8 Proposed RDNN model

This section describes the implantation details of the proposed RDNN model. This model has combined the features of the attention mechanism with CNN(AttCNN), attention with Bi-LSTM (AttBi-LSTM) and Hybrid Pooling(HPOOL) layers. The CNN with attention can give greater attention to the lexical features while processing the data. Each word in the output sentence maps the most relevant tokens from the input. In order to enhance the prediction accuracy, attention layers assign higher weights to these words. Each network

layer was based on the data set indicated by the position. If the position changes, the CNN's position invariant features will still detect which class the input belongs to. An LSTM, has focused on a certain part of the information in a particular time period only. AttBi-LSTM can encode exclusively focused semantic features of the words in the sentence at a particular time. so it can generate the most relevant features from the data. Max pooling can extract only maximum features whereas average pooling calculates the average features for each token in the feature vector. HPOOL layer combines the features of max and average pooling into a single feature vector. In order to make an efficient prediction, the model needs to be compiled with the best set of weights. The compiler entailed parameters namely loss function, optimizer, batch size and learning rate. Loss function is used to evaluate a set of weights and optimizer is used to seek out the different weights for the network. **Fig. 5.** depicts the experimental setup of the proposed model. In this work, the classifier model is assembled with categorical cross-entropy as a loss function, RMSprop acts as an optimizer, and categorical accuracy specified as metrics. The proposed model is superior to the CNN model, it enhances the weights to the most relevant features in the dataset.



**Fig. 5.** Neural Network model for rumor detection

The proposed RDNN algorithm is implemented using the following steps and neural network layers are modeled with the aid of sequential models. A feature vector is generated by cleaning the data corpus. Layers are arranged in the order to detect the rumor messages in the tweet. Each and every layer's hyper parameter is tuned based on the performance evaluated in the epoch.

#### Algorithm for Rumor Detection Neural Network

---

Input: [Training Data as Tr, Testing Data as Tt]

Output: [Predict Rumor keyword]

---

Step 1: Split training and testing data as 70:30

Step 2: for each term T in corpus  $C_n$  do

Generate vector matrix for each sentence in corpus

Calculate the maximum length of the vector

If  $(len\_mat <+ max\_length)$  then

Text\_padding =pad\_sequences(text\_data)

end for

Step 3: Build Attention Layer

Step 4: Call function for Attention Layer

Step 5: Output layer computation

Step 6: Attention layer configuration

Step 7: Model creation

Model<sub>seq</sub> -> Sequential()

Model<sub>seq</sub>.L1-> Conv1D( $\beta$ , kernal=3, $\sigma$ 'relu')

Model<sub>seq</sub>.L2-> BiDirectinalLSTM()

```

Modelseq.L3-> concatenate([avg_pool, max_pool])
Modelseq.L4->attention_in(Conv1D+Bi-LSTM+Pool)
Modelseq.L5->attention_out(attention_in)
Modelseq.L6-> Dense( $\sigma$ ='relu')
Modelseq.L7-> Dense( $\sigma$ ='softmax')
Modelseq.compile()

```

Data corpus is split into 70:30 for training and testing. A convolution layer is fed the optimal feature generated by the embedding layer. Attention layers receive the input. By applying the output of the convolution layer to Bi-LSTM with 1024 input parameters, optimal training parameters are selected for training. In the attention layers, the Bi-LSTM is processed with the training parameters. A sigmoid function is used as a classifier to categorize rumor (or) non-rumor messages.

## 5. Results and Discussion

### 5.1 Experimental results using machine learning approaches:

The preprocessed tweets are converted into the vector matrix with the aid of the Tf-Idf and CV vectorization techniques. It is used to determine the most influential terms in the dataset. In the TF-IDF feature, the context and words in twitter dataset are not arranged in order. The most frequently used terms have less importance than less frequently used terms in the feature vector which occupies large memory space. This generates a high volume of 0 vectors. The CV technique counts the number of words occurring in the tweet “gaja” dataset. The authors have calculated the vector model by the combination of the (1,2) i.e. unigram and bigram model. The experimental outcomes using vectorization methods applied on machine learning approaches like Naïve Bayes, SVM, Logistic Regression and Decision tree are depicted in **Table 7**. This shows the accuracy comparison among several machine learning algorithms in classifying rumors data and non-rumors data from “#gaja” and “Kaggle” twitter dataset. The rumor or non-rumor data is predicted based on the information provided by the ground truth. It is inferred that NB and SVM algorithm has achieved a high accuracy for #gaja dataset. NB algorithms predict the data based on the core assumption of all the features retrieved in the data. SVM finds a decision boundary by maximizing the distance between two classes. CV performs well on all machine learning algorithms. Because it extracts the highly flexible features from the dataset. The machine algorithms extract the hand crafted features, in order to improve the rumor classification task the dataset is modeled using deep learning models.

**Table 7.** Estimation of TF-IDF and CV for Gaja dataset using machine learning approaches

Machine Learning approaches	#gaja Dataset		Kaggle	
	TF-IDF	CV	TF-IDF	CV
Support Vector Machine	84.72	83.58	80.01	76.18
Logistic Regression	84.39	85.81	77.14	75.91
Multinomial Naïve Bayes	79.51	72.03	79.45	74.61
Decision Tree	62.92	62.43	79.45	74.61



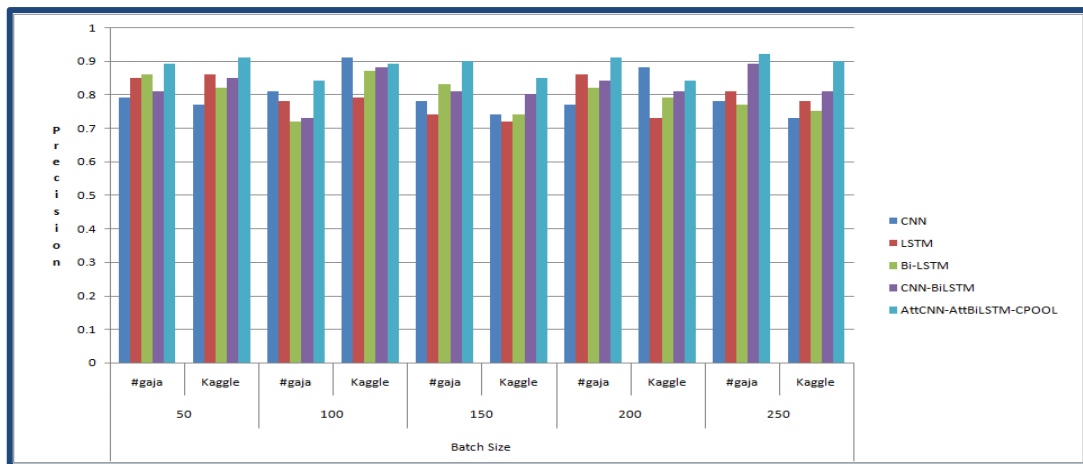
## 5.2 Experimental results using Deep Learning approaches:

Deep learning approaches were used in classifying tweet dataset as rumors and non-rumors data. The main extracted features specifically Glove and Fast Text used in both datasets. The accuracy is determined for evaluate performance in finding and classifying twitter dataset as normal and abnormal using proposed deep learning based model like CNN, LSTM, Bi-LSTM, CNN combined with Bi-LSTM, Att CNN integrated with Att Bi-LSTM and HPOOL shown in [Table 8](#). To estimate model accuracy, two kinds of metrics need to be found namely True positive (TP), and False Positive (FP). The valuation depends on a performance metric called precision. The precision is tagged by using formula (16).

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

### Influence of Batch Size

Initially, the authors have examined the influence of batch size of deep learning based Rumor Detection Neural Network model [15]. The batch size is a significant feature which manipulates variations in learning approaches. The model measured various dissimilar batch sizes between 50 and 250. The proposed model performance doesn't vary on several batch sizes as depicted in [Fig. 6](#).



**Fig. 6.** Correlation among Batch size and Precision

LSTM is extremely prospective to batch size. This confirms less performance while batch size varies. The truly positive predicted value among total outcomes is referred to as precision. Finally, the proposed RDNN method achieves better outcomes on various batch sizes.

### Influence of Learning Rate

The suitable preference of learning rate is highly significant to develop weight as well as offsets. Suppose, if the learning rate is excessively high, it is simple to surpass the tremendous position that formulates the system unbalanced. If the learning rate is tiny, automatically the training time is lengthy. The performance comparison of existing models with our proposed work is shown in [Fig. 7](#).

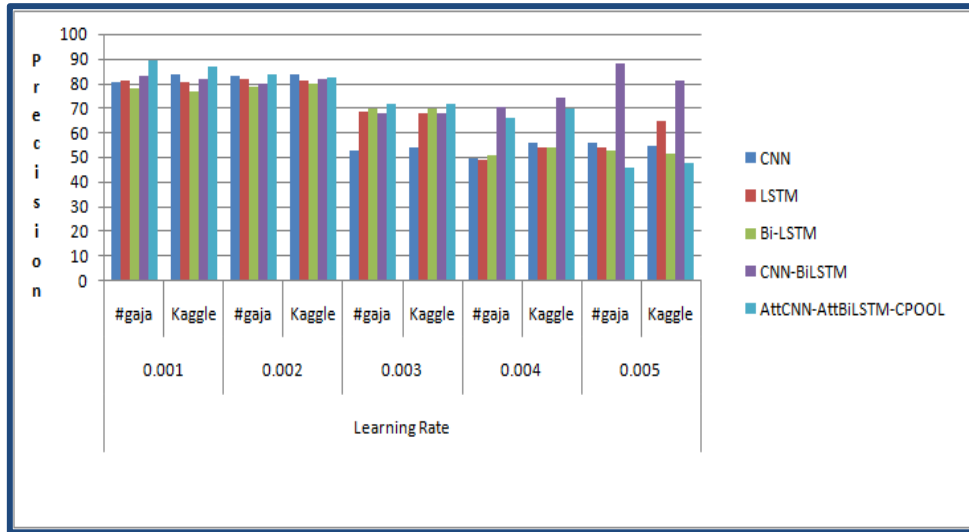


Fig. 7. Correlation among Learning Rate and Precision

LSTM and Bi-LSTM are very perceptible to dissimilar learning rates between 0.001 and 0.005. The RDNN model attains maximum outcome when compared with the learning rate of all baseline work.

**Influence of Filter Size**

Fig. 8 depicts that the proposed model attains greater precision as 92.4 % when size is set to 4, 5, and 6. We investigated various filter sizes namely [1, 2, 3], [2, 3, 4], [3, 4, 5], [4, 5, 6], and [5, 6, 7]. The novel RDNN model presents enhanced outcomes on all investigations. The evasion filter sizes utilized are set of 3, 4 and 5 in contrast with other models. The conventional CNN approaches produce further relative consequence with the RDNN model than other approaches.

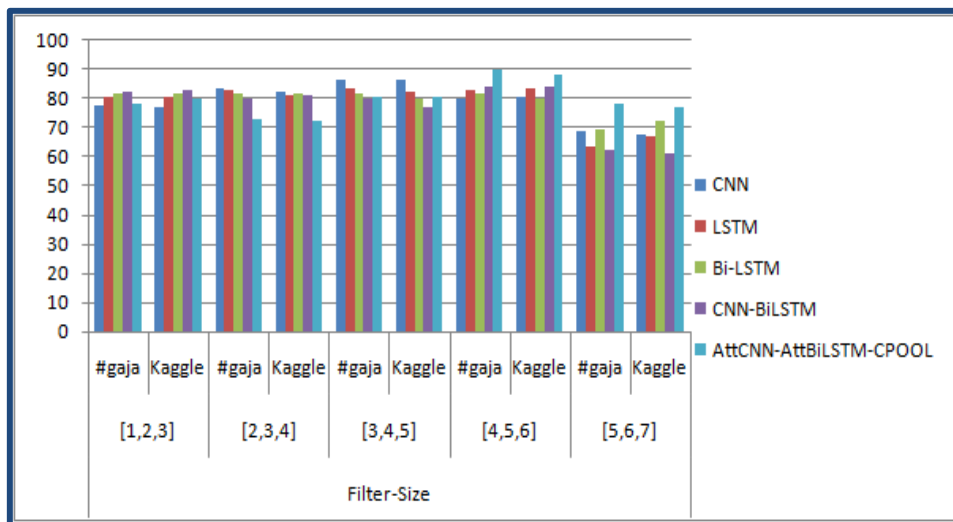


Fig. 8. Correlation among Filter Size and Precision

**Fig. 8** illustrates relative outcomes in terms of precision metrics, of every deep learning layers. It is obvious that the proposed RDNN presented improved results than other methods in CNN.

### 5.3 Investigational outcomes using deep learning approaches

The performance analysis of the RDNN model for the Gaja and Kaggle dataset based on the accuracy is depicted in **Table 8**. The proposed model is evaluated using two embedding techniques Glove and Fast Text. The proposed RDNN model has combined the features of the attention mechanism with CNN (AttCNN), attention with Bi-LSTM (AttBi-LSTM) and Hybrid Pooling(HPOOL) layers. The CNN with Attention has extracted the lexical level key features from the dataset and attention Bi-LSTM can encode the semantic features of the words in the whole sentence. HPOOL combines the down sampling of the input feature map from average and max-pooling layers.

**Table 8.** Accuracy estimation using deep learning models for Gaja and Kaggle datasets

Approaches used	Extracted Features	#gaja dataset accuracy (%)	Kaggle dataset accuracy (%)
CNN	Glove	81.65	80.73
	Fast Text	85.25	87.49
LSTM	Glove	82.10	89.45
	Fast Text	85.62	89.48
Bi-LSTM	Glove	81.91	86.44
	Fast Text	85.25	87.39
CNN + BiLSTM	Glove	89.94	93.96
	Fast Text	89.57	93.59
<b>RDNN</b>	<b>Fast Text</b>	<b>93.24</b>	<b>95.41</b>

From the accuracy perspective it has been observed that CNN (81.65%), LSTM(89.45%) and Bi-LSTM(86.44%) do not work well on glove embeddings. Fast Text embedding system achieves somewhat better accuracy of CNN (87.49%), LSTM(89.48%) and Bi-LSTM(87.39%) as compared to the Glove embedding. Because it can construct the word vector based on its character n-grams even if the word is not present in the training dataset. By combining the features of the CNN and Bi-LSTM, glove embedding performs well on the CNN+Bi-LSTM (93.96) and RDNN (93.22) deep learning models. The results exhibit that the performance of the classification model depends on the data corus and the type of the vectorization techniques. The experimental results illustrate that deep learning based Rumor Detection Neural Network achieves higher accuracy as 93.24% and 95.41% which enhances overall performance in categorizing rumors and non-rumors data. It is inferred that novel Rumor Detection Neural Network outperforms categorizing rumors as well as non-rumors tweets from tweet datasets.

## 6. Performance comparison of existing models

In the proposed Rumor Detection Neural Network the best three models are combined to detect the rumor messages are discussed in section 3. In **Table 9**, experimental results show

that the proposed model achieves 93.2%, 95.4% accuracy using FastText embedding techniques. Similarly the proposed model gives significant accuracy values for Glove embedding technique. The authors have also compared the training time of the model based on the vectorization method. It is inferred that the fastText technique is more efficient than the Glove embedding technique, because Fast Text embedding is to generate vectors from training data without using vocabulary. The proposed model is also compared with other existing rumor detection approaches as shown in **Table 9**. The proposed RDNN model outperforms by 93.2% and 95.41% using FastText embedding for Kaggle and gaja dataset.

**Table 9.** Performance Comparison of proposed system with existing models using Kaggle Dataset

Authors	Techniques	Text Features	Accuracy
Ahmed et al. (2017) [29]	Model for fake news detection using ML algorithm	TF-IDF and n-gram based TF	LR- 89% KNN- 83.1% SGD- 89% SVM- 86% LSVM-92% DT- 89%
Vivek Singh et al.(2017) [30]	Machine learning models based on well-established algorithms such as LR,SVM,RF,DT and kneighbors classifier	LIWC features	87%
Bali et al.(2019) [31]	Natural Language Processing with Machine Learning	TF-IDF	88%
<b>Proposed Model</b>	<b>RDNN</b>	<b>Glove</b>	<b>93.22</b>
		<b>Fast Text</b>	<b>95.41</b>

## 7. Conclusion

The goal of this research work is to analyze the best classifier model for detecting rumor messages posted in tweets for #gaja cyclone dataset and kaggle fake news dataset. The experiment is performed on these datasets using machine learning algorithms Support Vector Machine, Logistic Regression, Multinomial Naïve Bayes and Decision tree with TF-IDF and CV vectorization techniques. The support vector machine with TF-IDF has achieved 84.73% as its highest accuracy to classify the rumor tweets. Machine learning algorithms learn features from the data and make decisions based on the learning. It has a limited hyperparameter tuning capability and the prediction of new data is inappropriate. In order to make an efficient prediction of rumor messages the authors have proposed a novel Rumor Detection Neural Network to categorize rumor vs. non-rumor tweets. This model has integrated various neural network layers based on their features to retrieve better prediction of rumor tweets. In RDNN, its AttCNN layer is used to extract local and position invariant features from the data. An AttBi-LSTM layer to extract important semantic/contextual information and HPOOL combines the down sampling patches of the input feature maps from the average and maximum pooling layers. The RDNN model facilitates resolving the problem between efficiency and accuracy. RDNN achieves an accuracy of 95.41% for FastText and 93.24% for glove embedding techniques.

## References

- [1] Twitter Usage Statistics. [Online]. Available: <https://www.internetlivestats.com/twitter-statistics/>
- [2] Ahmad, S., Asghar, M. Z., Alotaibi, F. M., & Awan, I., "Detection and classification of social media-based extremist affiliations using sentiment analysis techniques," *Human-centric Computing and Information Sciences*, 9(1), 24, 2019. [Article \(CrossRef Link\)](#)
- [3] Alzanin S, Azmi A, "Detecting rumors in social media: a survey," *Proc. Comput Sci.*, 142, 294–300, 2018. [Article \(CrossRef Link\)](#)
- [4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by jointly learning to align and translate," in *Proc. of ICLR*, pp. 1-15, 2015. [Article \(CrossRef Link\)](#)
- [5] Caragea, Cornelia, Adrian Silvescu, and Andrea H. Tapia, "Identifying informative messages in disaster events using convolutional neural networks," in *Proc. of International conference on information systems for crisis response and management*, 2016. [Article \(CrossRef Link\)](#)
- [6] Chang C, Zhang Y, Szabo C, Sheng QZ, "Extreme user and political rumor detection on twitter," in *Proc. of International conference on advanced data mining and applications*, Springer, Cham, pp. 751–763, 2016. [Article \(CrossRef Link\)](#)
- [7] Chen, T., Li, X., Yin, H. and Zhang, J., "Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection," in *Proc. of Pacific-Asia conference on knowledge discovery and data mining*, pp. 40-52, June 2018. [Article \(CrossRef Link\)](#)
- [8] Graves, Alex, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2014. [Article \(CrossRef Link\)](#)
- [9] Guo, H., Cao, J., Zhang, Y., Guo, J. and Li, J., "Rumor detection with hierarchical social attention network," in *Proc. of the 27th ACM International Conference on Information and Knowledge Management*, pp. 943-951, October 2018. [Article \(CrossRef Link\)](#)
- [10] Han, H., Liu, J. and Liu, G., "Attention-based memory network for text sentiment classification," *IEEE Access*, vol. 6, pp.68302-68310, 2018. [Article \(CrossRef Link\)](#)
- [11] Jabreel, Mohammed, and Antonio Moreno, "A deep learning-based approach for multi-label emotion classification in tweets," *Applied Sciences*, 9(6), 1123, 2019. [Article \(CrossRef Link\)](#)
- [12] Kim, Yoon, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.
- [13] Liu, J., Yang, Y., Lv, S., Wang, J. and Chen, H., "Attention-based BiGRU-CNN for Chinese question classification," *Journal of Ambient Intelligence and Humanized Computing*, pp.1-12, 2019. [Article \(CrossRef Link\)](#)
- [14] Marianela García Lozanoa, Joel Brynielssona, Ulrik Frankec, Magnus Rosella, Edward Tjörnhammara, Stefan Vargab, Vladimir Vlassov, "Veracity assessment of online data," *Decision Support Systems*, 129, 113132, 2020. [Article \(CrossRef Link\)](#)
- [15] Muhammad Zubair Asghar, Ammara Habib, Anam Habib, Adil Khan, Rehman Ali, Asad Khattak, "Exploring deep neural networks for rumor detection," *Journal of Ambient Intelligence and Humanized Computing*, 12, 4315-4333, 2021. [Article \(CrossRef Link\)](#)
- [16] Nair, V., and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proc. of the 27th International Conference on Machine Learning*, Haifa, Israel, 807–814, 2010. [Article \(CrossRef Link\)](#)
- [17] Oluwaseun Ajao, Deepayan Bhowmik, Shahrzad Zargari, "Fake News Identification on Twitter with Hybrid CNN and RNN Models," in *Proc. of SMSociety*, pp. 226-230, July 2018. [Article \(CrossRef Link\)](#)
- [18] Prasannata Patwa, "Fake news, rumours on social media hit Indian firms,". [Online]. Available: <https://www.livemint.com/Companies/CqbmV2eOniYHZEqLYkxFyO/Fake-news-rumours-on-social-media-hit-Indian-firms.html>.
- [19] Severyn, Aliaksei, and Alessandro Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proc. of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 959-962, 2015. [Article \(CrossRef Link\)](#)

- [20] Veyseh APB, Ebrahimi J, Dou D, Lowd D, “A temporal Attentional model for rumor stance classification,” in *Proc. of the 2017 ACM on conference on information and knowledge management*, ACM, pp 2335–2338, 2017. [Article \(CrossRef Link\)](#)
- [21] Yubo, Chen, et al, “Event extraction via dynamic multi-pooling convolutional neural networks,” in *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 167–176, 2015. [Article \(CrossRef Link\)](#)
- [22] M Yu, Qunying Huang, Han Qina, Chris Scheele and Chaowei Yanga, “Deep learning for real-time social media text classification for situation awareness – using Hurricanes Sandy, Harvey, and Irma as case studies,” *International Journal Of Digital Earth 2019*, VOL. 12, NO. 11, pp. 1230–1247, 2019. [Article \(CrossRef Link\)](#)
- [23] Zeng, Y., Yang, H., Feng, Y., Wang, Z. and Zhao, D., “A convolution BiLSTM neural network model for Chinese event extraction,” *Natural Language Understanding and Intelligent Applications*, pp. 275-287, 2016. [Article \(CrossRef Link\)](#)
- [24] Zhang, Xiang, Junbo Zhao, and Yann LeCun, “Character-level convolutional networks for text classification,” *Advances in neural information processing systems*, 2015. [Article \(CrossRef Link\)](#)
- [25] Zhao Z, Wu Y, “Attention-based Convolutional neural networks for sentence classification,” in *Proc. of INTERSPEECH*, pp 705–709, 2016. [Article \(CrossRef Link\)](#)
- [26] SuthanthiraDevi, P., and Karthika, S., “Social Media Veracity Detection System Using Calibrate Classifier,” in *Proc. of International Conference on Computational Intelligence in Data Science*, pp. 85-98, February 2020. [Article \(CrossRef Link\)](#)
- [27] Deng, S., Huang, L., Xu, G., Wu, X. and Wu, Z., “On deep learning for trust-aware recommendations in social networks,” *IEEE transactions on neural networks and learning systems*, 28(5), pp.1164-1177, 2017. [Article \(CrossRef Link\)](#)
- [28] Gao, J., Han, S., Song, X. and Ciravegna, F., “RP-DNN: A Tweet level propagation context based deep neural networks for early rumor detection in Social Media,” *arXiv preprint arXiv:2002.12683*, 2020. [Article \(CrossRef Link\)](#)
- [29] Ahmed, H., Traore, I., & Saad, S., “Detection of online fake news using n-gram analysis and machine learning techniques,” in *Proc. of International conference on intelligent, secure, and dependable systems in distributed and cloud environments*, pp. 127-138, October 2017. [Article \(CrossRef Link\)](#)
- [30] Singh, V., Dasgupta, R., Sonagra, D., Raman, K. and Ghosh, I., “Automated fake news detection using linguistic analysis and machine learning,” in *Proc. of International conference on social computing, behavioral-cultural modeling, & prediction and behavior representation in modeling and simulation (SBP-BRiMS)*, pp. 1-3, 2017. [Article \(CrossRef Link\)](#)
- [31] Bali, A.P.S., Fernandes, M., Choubey, S. and Goel, M., “Comparative performance of machine learning algorithms for fake news detection,” in *Proc. of International Conference on Advances in Computing and Data Sciences*, pp. 420-430, April 2019. [Article \(CrossRef Link\)](#)
- [32] Kaggle data set at <https://www.kaggle.com/c/fake-news/data>
- [33] #gaja data set at [https://zenodo.org/record/4805619#.YK\\_JCaHhVPY](https://zenodo.org/record/4805619#.YK_JCaHhVPY)
- [34] Zhang, X., and Y. LeCun, “Text understanding from scratch,” 2015. [Article \(CrossRef Link\)](#)
- [35] Devi, P.S. and Karthika, S., “#CycloneGaja-rank based credibility analysis system in social media during the crisis,” *Procedia Computer Science*, vol. 165, pp.684-690, 2019. [Article \(CrossRef Link\)](#)





**P.SuthanthiraDevi** has received her B.E. degree in Computer Science Engineering from Madurai Kamaraj University and M.E. Degree in Computer Communication from Anna University. She is currently working towards her Ph.D. Degree in Anna University. Her research area is Social Network Analysis. She has 7 years of teaching experience at Anna University affiliated college. She has published nearly 14 research articles in refereed International Journals in areas like online social network and veracity analysis.



**Dr.S.Karthika** has received B.E (CSE) degree from Avinashilingam University and completed her M.E. (CSE) from Anna University. She has earned her Ph.D. degree under the Faculty of Information and Communication Engineering from Anna University. Her areas of research include key player analysis, text mining, and privacy in the online social network, rumour analysis, opinion mining, and Social IoT. To her credit, she has 12 refereed journals and 20 international conference publications.