Rumour Veracity estimation with Deep Learning for Twitter

Jyoti Prakash Singh[1], Nripendra P. Rana[2], Yogesh K. Dwivedi[2]

[1]National Institute of Technology Patna, Bihar, India
[2]School of Management, Swansea University Bay Campus, UK
jps@nitp.ac.in, nrananp@gmail.com and ykdwivedi@gmail.com

Abstract
Twitter has become a fertile ground for rumours as information can propagate to too many people in very short time. Rumours can create panic in public and hence timely detection and blocking of rumour information is urgently required. We proposed and compare machine learning classifiers with a deep learning model using Recurrent Neural Networks for classification of tweets into rumour and non-rumour classes. A total thirteen features based on tweet text and user characteristics were given as input to machine learning classifiers. Deep learning model was trained and tested with textual features and five user characteristic features. The findings indicate that our models perform much better than machine learning based models.
Keywords: Rumour veracity, deep learning, twitter, neural network, machine learning

1   Introduction

Social media has become essential part of our day-to-day life (Alalwan et al., 2017; Alryalat et al., 2017; Kuttimani et al., 2018; Shareef et al., 2019). Twitter (launched in July 2006) is currently one of the most popular social platforms that allows users to post any information, which is publicly visible. Each post on twitter is called tweet, which is limited in size to 280 characters. Due to short limits on tweets, users send and reads many tweets in a day. Information in the twitter diffuses very quickly through followers of a user. In a recent survey from Pew Research Centre, it is found that "two-third (67%) of Americans get news from social media". In the same survey it was also found that "about three-quarters (74%) of Twitter users get news on the site" (Shearer & Gottfried, 2017).

The major reason of having an up-to-date information on Twitter is the fact that anyone can instantly post, share, and gather information. Social media data are being used to develop systems for disasters (Kumar et al. 2019; Singh et al. 2017), location prediction of a user (Kumar et al., 2017), customer relationship management (Baabdullah et al., 2018; Kizgin et al., 2018; Kapoor et al., 2018; Shareef et al., 2019), and stock market prediction (Saumya et al., 2016). Unfortunately, the information cannot always be trusted upon. Some users post their tweets about events without any corroboration

and verification (Mendoza et al., 2010). DiFonzo & Bordia (2007) defined rumour as unverified and instrumentally relevant information statements in circulation. The open nature of Twitter is a suitable ground for rumourmongers to post and spread rumours. The rumour may result in major chaos and unpredictable reactions from involved individuals. An example of such a rumour is a tweet reporting an "Explosion at White House" in 2013. Only within three minutes, it created such a social panic that major stock indices (e.g. S&P 500 Index) dropped 14 points, and the Dow Jones Industrial Average also dropped about 145 points (Liu, Jin & Shen, 2018). Diffusion of negative information may cause fear and panic across people, disruption in social environments and affects the government credibility. To minimize the negative effects of rumour, it is essential to expose the falseness of information as early as possible before they can spread to larger extent.

A lot of research work across the world is being carried out to determine whether a tweet is rumour or not. In most of the literature, rumour detection is considered as a 2-class classification problem (Ma et al., 2015; Wu et al., 2015). Several features based on tweet text, propagation behaviour, user behaviour etc. are extracted to categorize a tweet into rumour and non-rumour class (Ma et al., 2015; Wu et al., 2015). The model performance of these systems depends on how accurately the features are extracted. Another group of researchers have tried to analyse the path through which the rumours are spreading on social media and used that information to differentiate rumour and non-rumour tweets (Kwon et al., 2017). The veracity or authenticity of the rumour information has also been used for tweet classification into rumour and non-rumour classes (Liang et al., 2015). The stance of the tweets whether a tweet is supporting a rumour, denying a rumour, questioning a rumour, or it is a normal comment is also being used to determine whether the tweet is rumour or non-rumour (Derczynski et al., 2017; Enayet & El-Beltagy, 2017).

In feature extraction and classification, deep learning architecture are now playing an important role. These architectures can be trained with a large amount of labelled data to learn the features directly from the data instead of extracting the features manually.

In this article, we have proposed and compared models based on machine learning as well as deep learning for rumour veracity determination. We extracted 13 features from tweet text and user characteristics to classify whether a tweet is rumour or not using different machine learning based classifiers. We compare the performance between machine learning and deep learning approaches and found that deep architecture performed better as compared to traditional machine learning approach. We utilized Long Short Term Memory (LSTM) network to exploit the deep representation of sequential data for rumour identification. Two proposed models based on the LSTM network are: (i) model using tweet text only, and (ii) model using tweet text along with user characteristics. By comparing both models, we found that textual features are sufficient to identify the rumour using an LSTM model. The main contributions of the research is proposal of deep learning based model for rumour veracity determination.

The rest of the paper is organized as follows: Section 2 discusses the related work; Section 3 represents the methodology of proposed work; Section 4 shows various experimental results. The results are discussed in Section 5 and the paper is concluded in Section 6.

2 Related Work

Determining the authenticity of information on social platform is a complex task. Identification of doubtful truth is very popular topic in social media. A number of researchers have worked for finding the truthfulness of information in this domain. Separating rumours information manually is not a trivial task. Hence, a number of supervised approaches have been proposed for automatic rumour identification. In this section, we present a brief description of some well-known work undertaken so far in this domain. There are two tasks where researchers have concentrated more i.e. i) stance classification where the researchers tried to determine the stance (type) of each tweet, and ii) veracity prediction where they determine the veracity or authenticity of rumour posts.

Oh et al. (2018) studied the acceptance of hate rumour and its consequence during a community crisis situation. They developed and tested a model using data collected from victims of a large scale (hate) rumour spread incident. Castillo et al. (2011) extracted several features, which are categorized as user-based, topic-based, content-based and propagation-based features to build a classifier. Qazvinian et al. (2011) developed a supervised approach for stance classification. In this stance, classification twitter specific rumour tweets are categorized as supporting rumour, denying rumour, questioning a rumour or neutral. They extracted several features from the time related information. According to Liang et al. (2015), the rumourmongers may have different behaviour from normal user and they investigate that replying the rumour post is different from the normal post. They proposed a user behaviour features based strategy and extracted eleven user behaviour features, which then treated as hidden representation for rumourmongers and possible rumour posts. They did their experiment with the Sina-Weibo data.

Zhao et al. (2015) explored the idea for early rumour identification. They used set of regular expressions to identify the questioning and denying tweets. Serrano et al. (2015) considered significance of time-span where they identified the difference of time between start of rumour spreading and start of anti-rumour spreading. Zubiaga et al. (2016) analysed how the rumours are diffusing on social media by utilizing the conversational posts. Using the various machine learning techniques, they explored the spreading of rumour on social media and evaluated how the users are supporting or denying a rumour to determine the veracity of a rumour. Lukasik et al. (2016) used Hawkes processes for modelling the diffusion of information on social platform in the context of stance classification. Hamidian & Diab (2016) explored the Tweet Latent Vector (TLV) approach. They proposed a TLV feature by applying the Semantic Textual Similarity (STS) model proposed by Guo and Diab (2012).

Jain et al. (2016) proposed an idea for identification of rumour source. They considered the network as undirected graph where whatever rumours are spreading on social network start from single source node with edges are connected to other nodes. We proposed a heuristic algorithm to get an estimate of the source node in as early as possible using the subgraph infected by rumour and the original graph of the network. Ma et al. (2017) developed an approach to identify rumours in social media posts using kernel learning method and propagation trees. They use the propagation tree for encoding the spread of the source tweet along with content, user and time associated with the retweeting nodes.

Zubiaga et al. (2017) explored the summarized survey of rumour detection on social media using different machine learning techniques. The survey discussed about the different approaches used for classification and detection of rumour and their performances. Srivastava et al. (2017) performed the stance classification as well as veracity prediction using cascading heuristics. By utilizing decision tree style, they trained a classifier with set of heuristics and then performance of the model is computed based on naive Bayes and winnow classifiers. Liu et al. (2017) considered the posts that had large amount of reposts and find out the difference between the rumour and non-rumour tree structure. They found that tree structure of rumour is deeper than the non-rumour and used that to separate rumour from non-rumour.

Previous approaches mostly focused on different features, which were derived from the linguistic information. However, in these approaches the performance of the system is reduced because they were limited to preserve the contextual meaning. Ma et al. (2016) proposed a recurrent neural network (RNN) based sequential approach for the veracity prediction. They counted three advantages over the traditional approaches. First, automatic feature learning capability. Second, reduction in the computation overhead. Third, capturing the semantics information. Chen et al. (2017b) explored a deep structure of neural network model, which is built on supervised classifier. They used the textual feature, which automatically learned by utilizing the different word representation techniques like pre-trained word embedding GloVe (Pennington et al., 2014).

They build a convolution neural network (CNN) for textual feature generation to preserve the contextual meaning and find the stance relationship using the source reply pairs of tweets. Rath et al. (2017) proposed a deep RNN architecture to identify the rumour spreaders by utilizing the trust. A user whose tweets are highly rated by other user is expected to have high trustworthiness score and a user who re-tweeted others tweet at high rate is expected to have high trust score and based on that score it is determined who can be possible rumour spreaders. Chen et al. (2017a) proposed a deep attention model based on recurrent neural network. Their model identifies the rumour by learning hidden sequential representation of posts. Chen et al. (2018) designed a model using recurrent neural network and auto encoder to learn the normal behaviour of individual users. They used the errors of different types of Weibo users to determine whether it is a rumour or not using self-adapting thresholds. They found that a two-layer model was performing better with an accuracy of 92.49% and F1 score of 89.16%. Lin et al. (2018) employed the LSTM and pooling operation of convolutional neural networks to build rumour identification models based on forwarding contents, spreaders and diffusion structures to detect rumours.

3   Method

3.1   Data

We have used the publically available dataset theme (Zubiaga et al., 2016) for our research work. The dataset contains rumour and non-rumour tweets and reactions on those tweets from five different events: Charlie Hebdo, Ferguson, German wings Crash, Ottawa Shooting, and Sydney Siege. Reaction is the collection of tweets replying to the source tweet. We have used 5802 source tweets, which include 1972 rumours and 3830 non-rumour tweets. The detail of the dataset is given in below in Table 1.

Table 1. Dataset with rumour and non-rumour

News	Rumor	Non-Rumor
Charlie Hebdo	22.0%	78.0%
Ferguson	24.8%	75.2%
German wings Crash	50.7%	49.3%
Ottawa Shooting	52.8%	47.2%
Sydney Siege	42.8%	57.2%

3.2   Models

We have proposed and evaluated three models for identifying rumour veracity. The first model was a traditional machine learning based model with supervised learning setting. We used four different classifiers to train and test the machine learning model. The classifier used are (i) Support Vector Machine, (ii) k-Nearest Neighbour, (iii) Gradient Boosting and (iv) Random Forest. The second model was deep learning based model with Long Short Term Memory (LSTM) network using tweet text only. The third and final model was also deep learning based model with LSTM network but it uses user metadata along with tweet text.

3.2.1. Model 1: Machine learning based model

We extracted thirteen different features from tweets contents and user characteristics to train and test the machine-learning model. The selected features are i) Existence of question ii) Detection of support words iii) Detection of denial words iv) Verified user or not v) Number of followers vi) Number of followees vii) Sentiment of tweet viii) Number of URL's ix) Number of hashtags x) User registered days xi) Length of tweet text xii) Status count xiii) Retweet count. The first three features are derived from the contents and rest of the features are extracted from user characteristics. The complete list of features with explanation along with their sources is shown in Table 2.

The rumour veracity determination problem is framed as a supervised learning task of classifying whether a tweet is rumour or not. Four different classifiers namely support vector machines; k-nearest neighbour, gradient boosting and random forest were used to evaluate the accuracy of the proposed model.

The major problem with the machine learning models is extracting the features from data, which is given as input to the model. In case of text oriented task, most of the extracted features ignore the sequential nature of a sentence. To mitigate this problem of manual features engineering and preserving the sequential text information, deep learning models are getting popularity.

Table 2. List of different features with type and their description

Feature	Description	Type
Question Existence	Whether the tweet contains question or not	Content Based
Support term detection	Extraction of support word like true, truth, exactly, possible, OMG, indeed etc.	
Denial term detection	Extraction of denial terms like not true, false impossible, don't agree, shut etc.	
Verify user	Whether the user is verified user or not	User Based
Number of followers	The number of users who follows an account	
Number of followees	The no of users who was followed by a post's author	
Sentiment of tweet text	Sentiment analysis decides the positivity or negativity of tweet text	
Number of URLs	The number of URL's in tweet text	
Number of Hashtags	The number of hashtags in tweet text	
User registered days	The number of days since user prole was created	
Length of tweet text	The length of the tweet text	
Status count	The total number of tweets posted by user	
Retweet count	The number of times a tweet is reposted	

3.2.2. Model 2: LSTM with Tweet Text only

Our second model was deep learning based model using LSTM network. The use of LSTM network preserves the contextual information of the text and also eliminates the need of hand-crafted features. The tweet text is embedded into a fixed size vector called embedding vector, which is given as input to the LSTM network in sequential manner. The schematic diagram of the model is given below in Figure 1.

Word embedding is a way to represent the words into a vector to preserve the co-occurrence information of words. To get the embedding vector, we started with a bag of all unique words from the tweet text used in the experiment. The tweet text was represented by one-hot encoding vector. A look-up matrix M is created to achieve the embedding of each word w_i. We used the pre-trained look up matrix GloVe available online on nlp.stanford.edu/projects/glove/. This pre-trained matrix is used between the input layer and first hidden layer of the network to create the word embedding. The look-up matrix M contains the vector of each word, which is represented as:

$$M = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1d} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2d} \\ \dots & & & & \\ w_{m1} & w_{m2} & w_{m3} & \dots & w_{md} \end{bmatrix} \qquad (1)$$
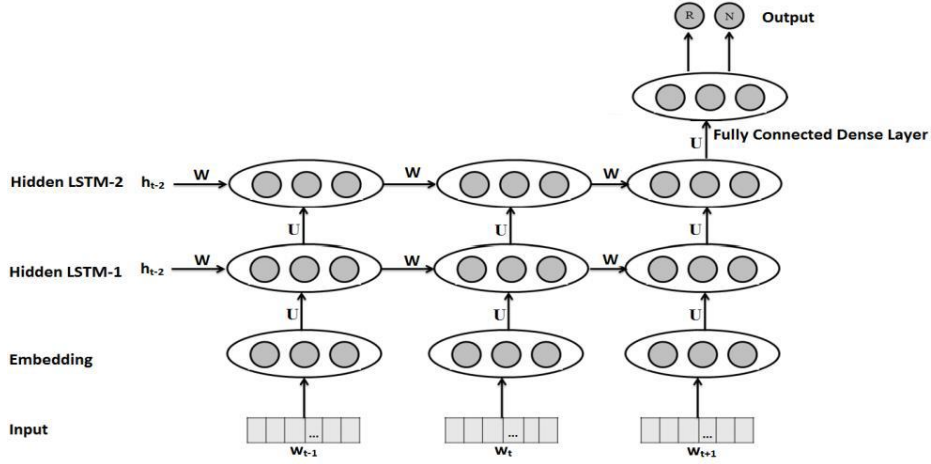

Fig. 1. Proposed LSTM module

Where, m represents total number of words and d represents the dimension to map these words. Embedding ($E(w_{1..m})$) can be represented as:

$$E(w_1..m) = e(w_1), e(w_2), e(w_3),..., e(w_m) \qquad (2)$$

Where $E(w_{1..m})$ denotes the embedding of all words present in a vocabulary and $e(w_1), e(w_2), e(w_3),..., e(w_m)$ denotes the embedding of a single word.

Our model was a 2-layer LSTM network where we used 2-hidden layers with 100 neurons at each LSTM unit. A summary of the network is shown in Figure 1. As can be seen from the figure, the input sentence length was fixed to 32 words. If the tweet has more than 32 words, then it was curtailed and only first 32 words are taken. On the other hand the short tweets are padded to make it of 32 words tweet. Each word was then embedded to a vector of length 200. The 200 length vector was scaled down to 100 length while passing through the first LSTM hidden unit. The length remains unchanged during processing by the second hidden LSTM unit. The 100 length output from LSTM unit is then passes through a dense layer, which scales it down to two outputs corresponding to rumour and non-rumour classes. In our experiment, we used 500 epochs with batch-size of 100. We used the 5-fold cross validation for training the system. The system was trained with four batch and validated with remaining one batch. We used the dropout as well recurrent-dropout at input connection and recurrent connection to the LSTM to prevent from over-fitting problem by randomly dropping some neurons in the training phase. At last, we used two neurons at dense layer for predicting

whether a tweet is rumour or not? The block diagram with input/output size is shown in Figure 2.
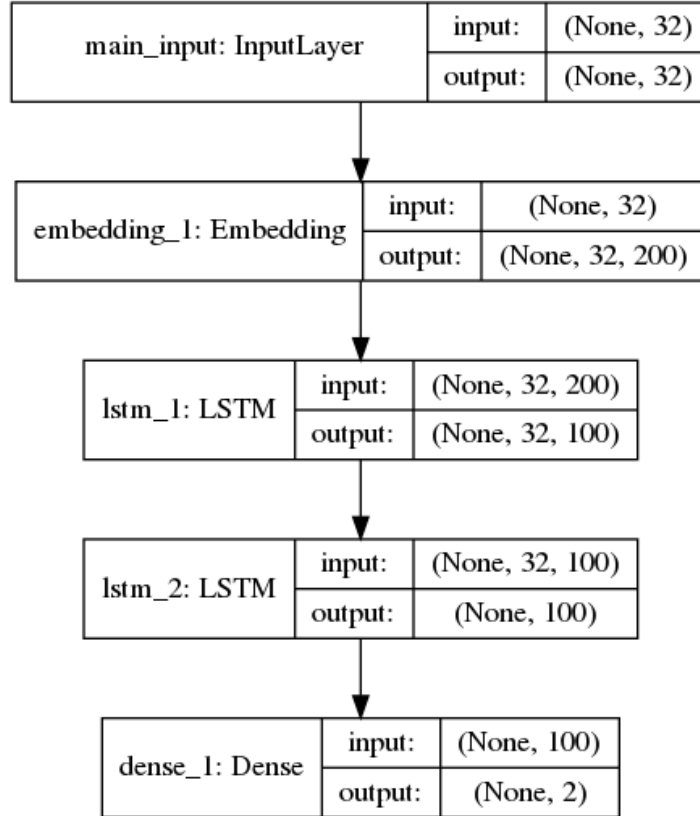
Fig. 2. LSTM with Tweet Text

3.2.3. Model 3: LSTM with Tweet Text and User Metadata

The user metadata like number of followers, number of followees, number of re-tweets, status count, verified user or not with the output of LSTM, which acts as input for the dense layer. Embedding is done as previous case, which is our third model is built by augmenting the second model to use the user information. We used a soft-max function to achieve non-linearity. At last we kept two neurons at dense layer, which is fully connected as our target was to identify whether the tweet is rumour or not.

4   Results

4.1   Result of machine learning model

We performed the experiment by extracting several features, which is given in Table 2 and use them to the four different classifiers i) SVM ii) Gradient Boosting iii) Random Forest iv) k-nearest Neighbour. We normalize our dataset using Gaussian normalization and split our dataset into 3:1 for training and testing purpose.

Table 3. Performance measure of different classifiers

Classifier	Class	Precision	Recall	F1-Score
SVM	0 (Non-rumour)	0.68	0.9	0.77
	1 (Rumour)	0.64	0.3	0.41
K-Nearest neighbour	0 (Non-rumour)	0.68	0.68	0.68
	1 (Rumour)	0.51	0.51	0.51
Gradient Boosting	0 (Non-rumour)	0.70	0.72	0.71
	1 (Rumour)	0.52	0.50	0.51
Random forest	**0 (Non-rumour)**	**0.72**	**0.89**	**0.79**
	1 (Rumour)	**0.70**	**0.42**	**0.53**

The result of SVM classifier is shown in Table 3, where our target class is class 1 or rumour and class 0 indicates non-rumour. Precision and Recall value for Class 1 were 0.64 and 0.30, which are quite low. The F1-Score for rumour class were 0.41. However, for Class 0 precision and recall and F1-score were 0.68, 0.90 and 0.77 respectively. The reason of higher performance matrices with Class 0 is the more number of data points of that class.

With k-nearest Neighbour the results are shown in Table 3. For Class 1 precision and recall value were 0.51 and 0.41 respectively. The F1-score value increased to 0.45. However, for Class 0 precision and recall and F1-score were 0.72, 0.89 and 0.79 respectively. Although there is slight increase in the performance metric, but, it is not of acceptable quality.

The result of the other two classifiers gradient boosting and random forest are tabulated in Table 3. The precision and recall value for rumour class is 0.52 and 0.50 respectively using gradient boosting. The F1-score for rumour class is improved to 0.51 and for random forest the precision and recall value for rumour class is 0.70 and 0.42. The F1-score for rumour class is improved to 0.53. These results are hardly better than a random classifier. We took these results as our benchmark to compare other two deep learning models.

4.2   Results of the LSTM network

The average precision, recall and F1-score values for rumour class are found to be 0.77, 0.69, and 0.72 respectively, which is shown in Table 4. As compared to our baseline (Random Forest) model average precision, recall and F1-score values using deep

learning approach improved by 7%, 27% and 19% respectively. We also tested our system performance by feeding text along with the user-metadata as an input but performance did not change.

Table 4. Precision, Recall, and F1-Score using 2-Layer LSTM Network Using Text

Class	Precision	Recall	F1-Score
0 (Non-rumour)	0.84	0.89	0.86
1 (Rumour)	0.77	0.69	**0.72**

5 Discussion

The major finding of the proposed research is that a 2-layer LSTM model using tweet text is performing better than machine learning based classifiers such as SVM, k-nearest neighbour, gradient boosting, and random forest. The LSTM model enhanced the performance by 35 % in terms of F1-score compared to the best performing machine learning classifier (random forest). The another major finding of the research is that the tweet text itself is a good predictor of whether a tweet is rumour or not as we compared two LSTM based model using text and other metadata along with text. The model with metadata such as followers count, followees count, status count and verified status did not reported any change in the performance metric because the number of metadata parameters were very low compared to the text size. The 2-layer LSTM based model is able to identify 84% of non-rumour tweet as non-rumour and 77% of rumour related tweets are classified as rumour related. The machine learning approach using 13 different features performed very badly with best recall value of 50% with gradient boosting algorithm. Based on the result of machine learning and LSTM based model, it can be concluded that the features used for machine learning models are not relevant as well as they not preserving the contextual meaning of the tweet text.

One major theoretical implication resulting from the research is that a system with 2 layer LSTM is preferred model for rumour detection from Twitter.

One of the practical implications of our proposed system is, it is capable of identifying the rumour as soon as possible with good accuracy, which can help to take appropriate decision by the government in the situation of public panic, disruption in social order, natural disaster and terrorist attack. The main limitation of our system is that it is language dependent model. It is tested for the dataset theme, which consist of only English language tweets. Our system may not perform well for scenarios where tweets are in code mixed language.

6 Conclusion

Determining veracity of the rumour whether the Tweet is a rumour or non-rumour is critical task on Twitter. In this study for early rumour identification we explored the idea of machine learning as well deep learning approach and compare their performances. The result of this study showed the effectiveness of deep learning over machine

learning regarding rumour identification. In machine learning approach we extracted several manual features from tweet contents and user meta-data information and evaluated their performance. The main limitation of the machine learning approach that it is very time-consuming process and limit to preserve the semantic representation of sequential information. On the other hand, to target the limitations of machine learning approach we used the deep learning module where our proposed LSTM model automatically learns the hidden temporal nature of tweet text which is difficult to preserve using hand-crafted features. Our deep architecture performed well over machine learning approach. The performance on detecting rumour veracity is not very high. Still accuracy of our proposed system can be improved. To know how deep learning can help in early rumour identification, more through experiments will be required. Due to overhead of data labelling, unsupervised approach can also be used.

7 References

1. Baabdullah, A. M., Rana, N. P., Alalwan, A. A., Algharabat, R., Kizgin, H., & Al-Weshah, G. A. (2018, June). Toward a Conceptual Model for Examining the Role of Social Media on Social Customer Relationship Management (SCRM) System. In *International Working Conference on Transfer and Diffusion of IT* (pp. 102-109). Springer, Cham.
2. Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. In Proceedings of the 20th international conference on World Wide Web (pp. 675{684). ACM.
3. Chen, T., Wu, L., Li, X., Zhang, J., Yin, H., & Wang, Y. (2017a). Call attention to rumours: Deep attention based recurrent neural networks for early rumour detection. arXiv preprint arXiv:1704.05973.
4. Chen, W., Zhang, Y., Yeo, C. K., Lau, C. T., & Lee, B. S. (2018). Unsupervised rumor detection based on users' behaviors using neural networks. *Pattern Recognition Letters*, *105*, 226-233.
5. Chen, Y.-C., Liu, Z.-Y., & Kao, H.-Y. (2017b). Ikm at semeval-2017 task 8: Convolutional neural networks for stance detection and rumour verification. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017) (pp. 465-469).
6. Derczynski, L., Bontcheva, K., Liakata, M., Procter, R., Hoi, G. W. S., & Zubiaga, A. (2017). Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. arXiv preprint arXiv:1704.05972.
7. Enayet, O., & El-Beltagy, S. R. (2017). Niletmrg at semeval-2017 task 8: Determining rumour and veracity support for rumours on twitter. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017) (pp. 470{474).
8. Hamidian, S., & Diab, M. (2016). Rumor identification and belief investigation on twitter. In Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (pp. 3-8).
9. Jain, A., Borkar, V., & Garg, D. (2016). Fast rumour source identification via random walks. Social Network Analysis and Mining, 6, 62.
10. Kumar, A., & Singh, J. P. (2019). Location reference identification from tweets during emergencies: A deep learning approach. *International Journal of Disaster Risk Reduction*, *33*, 365-375.
11. Kumar, A., Singh, J. P., & Rana, N. P. (2017). Authenticity of geo-location and place name in tweets.

12. Kwon, S., Cha, M., & Jung, K. (2017). Rumor detection over varying time windows. *PloS one*, *12*(1), e0168344.

13. Liang, G., He, W., Xu, C., Chen, L., & Zeng, J. (2015). Rumor identification in microblogging systems based on user's behaviour. IEEE Transactions on Computational Social Systems, 2, 99-108.

14. Liu, Y., Jin, X., & Shen, H. (2018). Towards early identification of online rumours based on long short-term memory networks. *Information Processing & Management*. doi.org/10.1016/j.ipm.2018.11.003

15. Liu, Y., Jin, X., Shen, H., & Cheng, X. (2017). Do rumours diffuse differently from non-rumors? A systematically empirical analysis in sina weibo for rumour identification. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 407-420). Springer.

16. Lukasik, M., Srijith, P., Vu, D., Bontcheva, K., Zubiaga, A., & Cohn, T. (2016). Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 393¬398) volume 2.

17. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K.-F., & Cha, M. (2016). Detecting rumours from microblogs with recurrent neural networks. In IJCAI (pp. 3818-3824).

18. Ma, J., Gao, W.,Wei, Z., Lu, Y., &Wong, K.-F. (2015). Detect rumours using time series of social context information on microblogging websites. In Proceedings of the 24th ACM International on Conference on Information and knowledge Management (pp. 1751-1754). ACM.

19. Ma, J., Gao, W., & Wong, K.-F. (2017). Detect rumours in microblog posts using propagation structure via kernel learning. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 708-717) volume 1.

20. Oh, O., Gupta, P., Agrawal, M., & Rao, H. R. (2018). ICT mediated rumour beliefs and resulting user actions during a community crisis. *Government Information Quarterly*, *35*(2), 243-258.

21. Qazvinian, V., Rosengren, E., Radev, D. R., & Mei, Q. (2011). Rumor has it: Identifying misinformation in microblogs. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 1589-1599). Association for Computational Linguistics.

22. Rath, B., Gao, W., Ma, J., & Srivastava, J. (2017). From retweet to believability: Utilizing trust to identify rumour spreaders on twitter. In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017 (pp. 179-186). ACM.

23. Saumya, S., Singh, J. P., & Kumar, P. (2016, September). Predicting stock movements using social network. In *Conference on e-Business, e-Services and e-Society* (pp. 567-572). Springer, Cham.

24. Serrano, E., Iglesias, C. A., & Garijo, M. (2015). A survey of twitter rumour spreading simulations. In Computational Collective Intelligence (pp. 113-122). Springer.

25. Shearer, E., & Gottfried, J. (2017). News use across social media platforms 2017. *Pew Research Center*, *7*.

26. Srivastava, A., Rehm, G., & Schneider, J. M. (2017). Dfki-dkt at semeval-2017 task 8: rumour detection and classification using cascading heuristics. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017) (pp. 486¬490).

27. Wu, K., Yang, S., & Zhu, K. Q. (2015). False rumours detection on sina weibo by propagation structures. In Data Engineering (ICDE), 2015 IEEE 31st International Conference on (pp. 651-662). IEEE.

28. Zhao, Z., Resnick, P., & Mei, Q. (2015). Enquiring minds: Early detection of rumours in social media from enquiry posts. In Proceedings of the 24th International Conference on

World Wide Web (pp. 1395-1405). International World Wide Web Conferences Steering Committee.

29. Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. (2017). Detection and resolution of rumours in social media: A survey. arXiv preprint arXiv:1704.00656.

30. Zubiaga, A., Liakata, M., Procter, R., Hoi, G. W. S., & Tolmie, P. (2016). Analysing how people orient to and spread rumours in social media by looking at conversational threads. PloS one, 11.

31. Shareef, M.A, Mukerji, B., Dwivedi, Y.K., Rana, N.P. & Islam, R. (2019). Social Media Marketing: Comparative Effect of Advertising Sources. *Journal of Retailing and Consumer Services*, 46, 58-69.

32. Roy, P.K., Singh, J.P., Baabdullah, A., Kizgin, H., & Rana, N.P. (2018). Identifying Reputation Collectors in Community Question Answering (CQA) sites: An Exploration of the Dark Side of Social Media. *International Journal of Information Management*, 42, 25-35.

33. Tamilmani, K., Rana, N.P., Alryalat, M., Alkuwaiter, W., & Dwivedi, Y.K. (2018). Social Media Research in the Context of Emerging Markets: An Analysis of Literature Published in Senior Scholars' Basket of IS Journals. *Journal of Advances in Management Research*, DoI: 10.1108/JAMR-05-2017-0061.

34. Kizgin, H., Jamal, A., Dey, B., and Rana, N.P. (2018). The Impact of Social Media on Consumers' Acculturation and Purchase Intentions. *Information Systems Frontiers,* 20(3), 503-514.

35. Kapoor, K.K., Tamilmani, K., Rana, N.P., Patil, P., Dwivedi, Y.K. & Nerur, S. (2018). Advances in Social Media Research: Past, Present and Future. *Information Systems Frontiers*, 20(3), 531-558.

36. Alryalat, M., Rana, N.P., Sahu, G.P., Dwivedi, Y.K. and Tajvidi, M. (2017). Use of Social Media in Citizen-Centric Electronic Government Services: A Literature Analysis. *International Journal of Electronic Government Research*, 13(3), 55-79.

37. Alalwan, A.A., Rana, N.P., and Dwivedi, Y.K., & Algharabat, R.S. (2017). Social Media in Marketing: A Review and Analysis of the Existing Literature. *Telematics and Informatics,* 34(7), 1177-1190.