Fake News Detection in Hindi Using Embedding Techniques

Pasi Shailendra, Rashmi M, Ramu S, Ram Mohana Reddy Guddeti
Department of Information Technology
National Institute of Technology Karnataka, Surathkal, India
{pasishailendra491, nm.rashmi, ramu68}@gmail.com, profgrmreddy@nitk.edu.in

2022 IEEE Region 10 Symposium (TENSYMP) | 978-1-6654-6658-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/TENSYMP54529.2022.9864378

***Abstract*—Internet users have been rapidly increasing in recent years, especially in India. That is why nearly everything operates in an online mode. Sharing information has also become simple and easy due to the internet and social media. Almost everyone now shares news in the community without even considering the source of information. As a result, there is the issue of disseminating false, misleading, or fabricated data. Detecting fake news is a challenging task because it is presented in such a form that it looks like authentic information. This problem becomes more challenging when it comes to local languages. This paper discusses several deep learning models that utilize LSTM, BiLSTM, CNN+LSTM, and CNN+BiLSTM. On the Hostility detection dataset in Hindi, these models use Word2Vec, IndicNLP fastText, and Facebook's fastText embeddings for fake news detection. The proposed CNN+BiLSTM model with Facebook's fastText embedding achieved an F1-score of 75%, outperforming the baseline model. Additionally, the BiLSTM using Facebook's fastText outperforms CNN+BiLSTM using Facebook's fastText on the F1-score.**

***Index Terms*—Natural Language Processing, Fake news, Embedding, Deep learning models, Hindi language.**

I. Introduction

With the rapid growth of Internet users, nearly all services are now available online. Almost every age group has internet access. Previously, individuals had to wait for newspapers, radio news, and television channels to obtain information and news about any domain. However, one can now access real-time news with a single click and share it with a group of people or a community. This ease in information sharing service does not come without some drawbacks. This can be done for financial gain, create a biased opinion about someone, or incite hatred between religions.

Fake news is a piece of information that can misguide or provide wrong information to the user/reader. This can be against an individual or against an organization, community, etc., which can change an individual's or society's image in a wrong way. This type of news spread very fast. Fake news can cause a riot, hate, financial loss, etc. Even well-established news-providing agencies can not handle fake news.

Natural Language Processing (NLP) is a branch of Artificial Intelligence (AI) that gives machines the ability to understand human languages. It is a discipline that works on the interaction between machines and human languages [1]. NLP is booming nowadays in various industries and is also being implemented in other domains. The classical application of NLP is Information Retrieval, Question-Answering, Machine Translation, and Sentiment analysis.

This research aims to stop the spread of this misguiding information, which can harm an entire community with just a single wrong piece of information. This misleading information is responsible for riots, mob lynching, character assassination, and other dangerous activities [2]. There are many news agencies, and most agencies are not fact-checking on the received information and publishing it to gain readers. Most people check their news feed on their phone at regular intervals, and when they encounter some breaking news, they share it immediately. This news can also be misleading fake news, spreading into society rapidly.

Detecting fake news is complex since it is presented in a form that seems to be a fact, and readers rarely cross-check this news. The problem becomes more challenging while detecting fake news in local languages. Because for the English language, there are rich resources available with which the task becomes somewhat easy to solve. But there are very few resources for local languages, and that information is not also authentic [3]. That is why we have used some of the pretrained resources related to the Hindi language for the fake news detection task.

Word embedding represents different words having the same meaning with similar real-valued vector representation. In this method, each term in the document is defined in high dimensional space by a real-valued vector. We have used Word2Vec word embedding [4], IndicNLP fastText [5], and Facebook's fastText embedding [6]–[11]. Authors in [12] proposed the n-gram approach using machine learning algorithms, but the n-gram approach doesn't work with unseen instances in its feature space. Long Short Term Memory (LSTM) [13] is used in this work to remember the long-term output of previous layers. CNN layer is used before the LSTM and BiLSTM models to extract the contextual features in CNN+LSTM and CNN+BiLSTM model, respectively. The Key contributions in the proposed work are the following:

- Word embedding generation for each token using Word2Vec, Facebook's fastText, and IndicNLP's fastText and compared these embeddings.
- Experiments were conducted using various LSTM-based deep learning models.

- Comparison of LSTM, BiLSTM, CNN+LSTM, and CNN+BiLSTM using Word2Vec, Facebook's fastText, and IndicNLP's fastText embeddings based on their F1-score.

The remaining sections of this paper are as follows. Various works on fake news detection are discussed in Section II. Section III elaborates on proposed method. Section IV details the various experiments conducted. In Section V, we bring this study to a conclusion.

II. RELATED WORK

Numerous studies on the detection of fake news have been conducted in recent years. But, the majority of these works are based on English language. Some of the works relate to detecting fake news, and some are related to analyzing fake news propagation in social media. Mostly fake news detection task focuses on social networks like Twitter/Facebook [14]. The study by Ahmed *et al*. [2] shows extracting the linguistic features from text articles with n-gram and then training the machine learning models with KNN (K-nearest neighbor), SVM (Support vector machine), LR (Logistic Regression), Decision Tree, and SGD (Stochastic Gradient Descent) where SVM and LR models performed the best. Wang *et al*. [15] introduced Event Adversarial Neural Network based method for detecting the fake news using Twitter [16] and Weibo [17] dataset. Ahmad *et al*. [18]] developed ensemble method to machine learning models. Kamal *et al*. [19] implemented "Hostility detection in the Hindi language" using transformer-based Hindi language models. [20] used translated datasets in the Hindi language to evaluate models on CNN, LSTM, and Attention, Multilingual pre-trained embeddings based on BERT[21]. Baruah *et al*. [22] elaborated the method that is used to identify fake news spreaders. They applied the pre-trained large cased BERT model to make the classification. They concatenated all the tweets of an author and then performed classification. The tweets of each of the authors were processed and evaluated separately. Babu *et al*. [23] proposed machine learning techniques for fake news detection with the Internet of Things. Monther Aldwairi and Ali Alwahedi [24] presented a tool based on a database that stores the fake news dataset with their sources. The tool predicts these sources based on the database information and blocks these sources from the Web. Islam *et al* [25]. Proposed the LSTM model with word Embedding of the dataset for detecting rumors about covid. But the model suffers from the skewness of data.

III. METHODOLOGY

Initially, dataset cleaning and preprocessing are performed on the dataset. After that, feature extraction is done on the dataset, which is used for the training of the models. Various embedding techniques are used for feature extraction like Word2Vec embedding, IndicNLP fastText embedding, and Facebook fastText embedding. Simple LSTM, BiLSTM, and also Hybrid model is implemented with CNN and LSTM and also with BiLSTM models. Various components in the proposed method is depicted in Fig. 1.

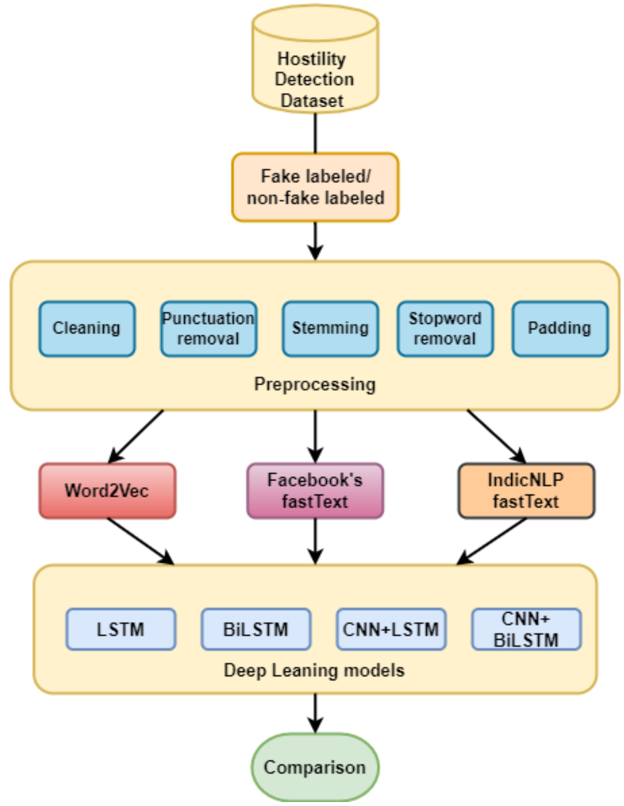

Fig. 1. Overall architecture of proposed method.

A. Preprocessing

Using a dataset directly for feature extraction is ineffective because it contains a large amount of irrelevant data. That is why cleaning and preprocessing are required. In data cleaning, irrelevant details such as URLs, numbers, and emojis are filtered out. Punctuation removal, stemming, stop-word removal and tokenization are all performed during data preprocessing. Because no two posts are identical in length, sequence padding is used to ensure that the input is uniform in length. From Fig. 2, we can see that very few posts have a length of more than 120 words. That is why we have considered the maximum length of the post as 120. Posts with a length of more than 120 are neglected, and for posts with a length lesser than max length, sequence padding is used.

B. Feature Extraction

Text data can not be directly fed to models because models can only understand the data in numerical format. There are different techniques to perform feature extraction. That is why there is a need to convert these inputs into a numerical format, or we can say into vector format, which is easily understood by the models. Converting these text/images into vectors is also called feature extraction. Word2Vec embedding, IndicNLP fastText embedding, and Facebook fastText embedding are used as a feature.

1) Word2Vec: Basic feature extraction methods like Tf-IDF, Bag of words, etc., are not able to get the semantic meaning from the text document. There comes the Word2Vec for feature extraction. Word2Vec model gives the semantic relationships between words. In Word2Vec, similar words will have mostly similar vector presentations. For example. Orange and apple will have identical vector representations, whereas a dog will have very much different vector representations than orange and apple. Word2Vec model learns semantics with the neighboring words. The input to the Word2Vec is given in the form of a corpus. We have used our dataset as a corpus and used it for training Word2Vec model. Output of Word2Vec model is a feature vector that shows the words of that corpus.

2) IndicNLP fastText Embedding: Indic fastText model is particularly developed for Indian languages by AI4Bharat. Indic fastText model is available in 11 different Indian languages. We have used a Hindi pretrained model with a dimension of 300.

3) Facebook's fastText Embedding: The fastText library is developed by Facebook's Artificial Intelligence Research Lab for learning word embeddings. Vector representation of words can be obtained by using this model, which is trained in 157 different languages. We have used pre-trained word embedding of Hindi language with a dimension of 300, n-gram length of 5 trained with CBOW method.

C. Model Architecture

We have used deep learning models with various embedding techniques. LSTM, BiLSTM, and also CNN with LSTM, CNN with BiLSTM models are implemented, which are as follows:

1) LSTM: The model is built sequentially with an Embedding layer, a single LSTM layer of 32 units, a dense layer of size 32 with ReLU activation, dropout (0.4), and a dense layer of size 1 with sigmoid activation.

2) BiLSTM: This sequential model is built with Embedding layer followed by a dense layer of size 32 with ReLU activation, 4 bidirectional LSTM layer with return sequence with 64 units, 1 bidirectional LSTM with 32 units, dense layer of size 32 with ReLU activation, dropout (0.3), dense layer of size 1 along with sigmoid activation.

3) CNN+LSTM: This model is built sequentially with Embedding layer followed by 1D CNN with filter size 64 and kernel size 4, LSTM with 64 units, dropout (0.5), LSTM with 64 units, two dense layers of size 50 with ReLU activation, dense layer of size 64 with l2 regularizer and dense layer of size 1 along with sigmoid activation.

4) CNN+BiLSTM: This model is built similar to the CNN+LSTM model, except the LSTM layers are replaced by Bidirectional LSTM layers and 1D CNN of filter size 128. Adam optimizer is used as an optimizer, and while training, early stopping is used to handle the overfitting of the model.

IV. Experiments, Results, and Analysis

Series of experiments were conducted to explore the efficiency of combination of various LSTM and embeding methods in detecting the fake news. We have considered all the basic evaluation metrics such as Accuracy, Precision, Recall, and F1-score.

A. Dataset

Hostility detection dataset in Hindi is used in the paper, which is a part of the CONSTRAINT2021 shared task (con 2021) [26]. It is a collection of 8200 online posts. The dataset provides a total of 1638 fake, 1132 hate, 1071 offensive, 810 defame, and 4358 non-hostile labeled posts. Each post is multi-labeled. We have considered all posts with a fake label as fake labeled and non-hostile posts that are not labeled as fake to non-fake label. There is a total of 1638 fake and 2196 non-fake news available in the dataset. Most of the posts are not exceeding the length of 120 words in the dataset.

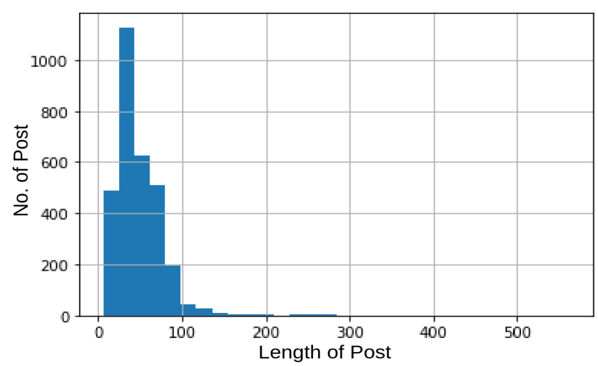

Fig. 2. Length of the Posts in dataset

We have experimented with LSTM, BiLSTM, CNN+LSTM, and CNN+BiLSTM model with Word2vec embedding, which is trained on our local corpus built with Hostility dataset. The results with Word2Vec word embedding are shown in Table 1 and Fig. 3. As this embedding is trained with local corpus, it doesn't have a rich dictionary for word vector representation. Performance with Word2Vec embedding is slightly lesser than the baseline model. The Baseline model is implemented with Logistic regression using pretrained m-BERT sentence embedding.

Models have experimented with IndicNLP fastText embedding also. Indic fastText embedding is less effective than the word2Vec embedding, as shown in Table 2 and Fig. 4. IndicNLP fastText models are trained with limited corpora, so it is ineffective for fake news detection on the Hostility dataset.

Facebook's fastText models are developed by Facebook AI Research Laboratory, which is trained on Wiki data which is a very large corpus. That is why it is very much effective on fake news detection in Hindi. Results of Facebook's fastText embedding with deep learning models are reported in Table 3 and Fig. 5.

TABLE 1
PERFORMANCE OF DEEP LEARNING MODELS USING WORD2VEC EMBEDDING.

Model	Accuracy	Precision	Recall	F1-score
LSTM	61%	60%	62%	61%
BiLSTM	63%	64%	62%	63%
CNN+LSTM	61%	60%	62%	61%
CNN+BiLSTM	66%	68%	64%	66%

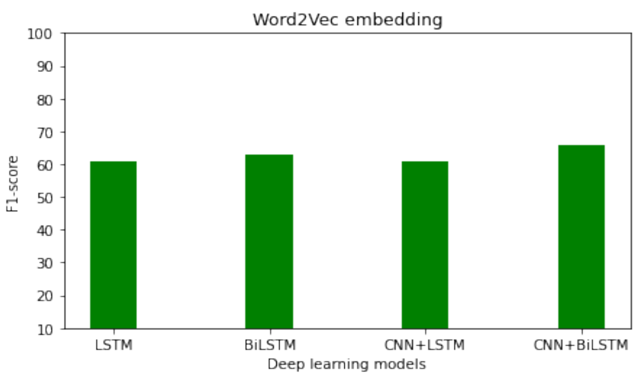

Fig. 3. Performance of Word2Vec embedding

TABLE 2
PERFORMANCE OF DEEP LEARNING MODELS USING INDICNLP FASTTEXT EMBEDDING.

Model	Accuracy	Precision	Recall	F1-score
LSTM	61%	60%	54.2%	57%
BiLSTM	59%	56%	56%	56%
CNN+LSTM	58%	58%	56%	57%
CNN+BiLSTM	60%	53.5%	61 %	57%

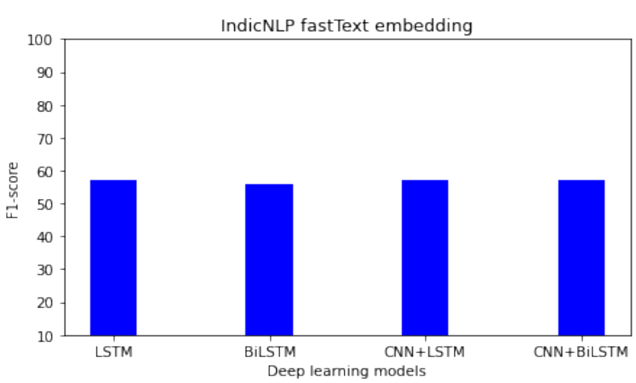

Fig. 4. Performance of IndicNLP fastText embedding

TABLE 3
PERFORMANCE OF DEEP LEARNING MODELS USING FACEBOOK'S FASTTEXT EMBEDDING.

Model	Accuracy	Precision	Recall	F1-score
LSTM	75%	74.5%	73.5%	74%
BiLSTM	75%	75%	75%	75%
CNN+LSTM	75%	76%	71.2%	73.5%
CNN+BiLSTM	76%	76%	74%	75%

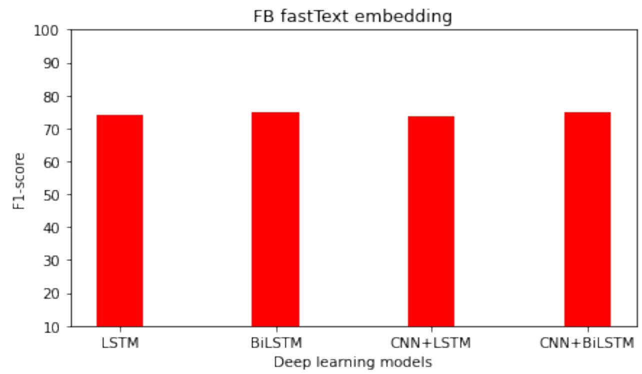

Fig. 5. Performance of Facebook's fastText embedding

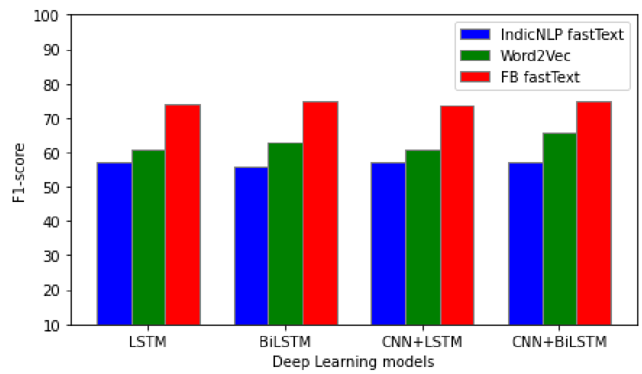

Fig. 6. Comparison of DL models with different embeddings

The baseline model discussed in the dataset paper has the F1-score of 68% with a Logistic regression model using m-BERT embedding. We got the F1-score of 75% with BiLSTM and CNN+BiLSTM using Facebook's fastText embedding. A comparison of the Proposed model performance with the baseline model and the state-of-the-art result is shown in Table 4. The Indic-BERT model [19] performs slightly better than the proposed model on F1-score, but the Indic-BERT model is much more complex than the proposed CNN+BiLSTM model. The Indic-BERT model has 31 million parameters, whereas the proposed CNN+BiLSTM model has only 4 million parameters. Even though both BiLSTM and CNN+BiLSTM give the F1-score of 75% with Facebook's fastText embedding, if we consider accuracy also, then CNN+BiLSTM performs better than other models. The overall comparison of different models with all three embedding techniques is shown in Fig. 6. The output of the CNN+BiLSTM model on a few sample posts using Facebook's fastText embedding is shown in Fig. 7 with actual labels (ground truth).

Posts	Predicted label	Actual label
बॉयकॉट चाइना लिखे हुए प्रोडक्ट्स का उत्पादन खुद चीन में ही किया जा रहा है।	1	1
देश का 3000 करोड़ रुपया बह गया पानी मे।' 'देश का 3000 करोड़ रुपया बह गया पानी मे जिस देश मे करीबन 1 करोड़ लोग रोज भूखे सोते हैं बेगैर इलाज के रोज लाखों मर रहे हैं उस देश मे इतनी क़ीमती मूर्ती शोभा नहीं देता	0	1
यदि आप जातिवादी सोच से ग्रसित हैं तो आप राष्ट्रवादी हो ही नहीं सकते, हिन्दूराष्ट्र-निर्माण की सबसे बड़ी रूकावट जातिवाद है..	0	0
कोविड के वक़्त बेहतरीन ख़बर - राष्ट्रपति भवन के अंदर स्थित मुग़ल गार्डन का नाम बदल कर डॉ राजेंद्र प्रसाद गार्डन कर दिया गया है |	1	1
जब सरकार हिटलर शाही हो जाए और शैक्षिणिक संस्थानों को जालिया वाला बाग बना दे, धर्मों की आज़ादी छीन ले, बोलने की आज़ादी छीन ले ,मांगों की आज़ादी छीन ले,सरकार के खिलाफ आवाज़ उठाने की आज़ादी छीन ले, मीडिया की आज़ादी छीन ले ,तब दल्ले पत्रकार आज़ादी मांगी नही छीन कर ही ली जाती है।	1	0

Fig. 7. Sample output of proposed model

TABLE 4
COMPARISON OF PROPOSED MODEL WITH BASELINE AND STATE-OF-THE-ART MODEL.

Model	F1-score
Baseline model (m-BERT with LR) [26]	68%
Proposed CNN+BiLSTM	**75%**
Indic-BERT [19]	77%

V. CONCLUSION

This paper proposed deep learning models for detecting fake news in the Hindi language using Word2Vec, IndicNLP fastText, and Facebook's fastText embedding on the Hostility detection dataset. Basic LSTM and BiLSTM models are used with different embeddings. Also, a combination of CNN, LSTM, and BiLSTM is implemented with Word2Vec, IndicNLP fastText, and Facebook's fastText embedding, where Facebook's fastText embedding resulted in the best results. So, we can conclude that the model does not affect that much, but the knowledge/ embedding given to the model matters the most. Having an embedding trained on a large corpus gives better results. In this work, BiLSTM and CNN+BiLSTM obtained better results using Facebook's fastText embedding than the baseline systems with Logistic regression using m-BERT embedding.

In the future, we will investigate various Transformer-based language models pre-trained on Indian languages in attempt to improve performance.

REFERENCES

[1] J. F. Allen, "Natural language processing," in *Encyclopedia of computer science*, 2003, pp. 1218–1222.

[2] Y. Tsfati, H. G. Boomgaarden, J. Strömbäck, R. Vliegenthart, A. Damstra, and E. Lindgren, "Causes and consequences of mainstream media dissemination of fake news: literature review and synthesis," *Annals of the International Communication Association*, vol. 44, no. 2, pp. 157–173, 2020.

[3] S. Singhal, R. R. Shah, and P. Kumaraguru, "Factorization of fact-checks for low resource indian languages," *arXiv preprint arXiv:2102.11276*, 2021.

[4] K. W. Church, "Word2vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.

[5] D. Kakwani, A. Kunchukuttan, S. Golla, N. Gokul, A. Bhattacharyya, M. M. Khapra, and P. Kumar, "Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 4948–4961.

[6] I. Facebook, *fastText: Library for fast text representation and classification*, 2016. [Online]. Available: https://github.com/facebookresearch/fastText

[7] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," *arXiv preprint arXiv:1802.06893*, 2018.

[8] L. Mouselimis, *fastText: Efficient Learning of Word Representations and Sentence Classification using R*, 2021, r package version 1.0.1. [Online]. Available: https://CRAN.R-project.org/package=fastText

[9] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017, pp. 427–431.

[10] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, M. Douze, and H. Jegou, "Fasttext.zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[11] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[12] H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques," in *International conference on intelligent, secure, and dependable systems in distributed and cloud environments*. Springer, 2017, pp. 127–138.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of economic perspectives*, vol. 31, no. 2, pp. 211–36, 2017.

[15] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. Gao, "Eann: Event adversarial neural networks for multi-modal fake news detection," in *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, 2018, pp. 849–857.

[16] C. Boididou, K. Andreadou, S. Papadopoulos, D.-T. Dang-Nguyen, G. Boato, M. Riegler, Y. Kompatsiaris *et al.*, "Verifying multimedia use at mediaeval 2015." *MediaEval*, vol. 3, no. 3, p. 7, 2015.

[17] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo, "Multimodal fusion with recurrent neural networks for rumor detection on microblogs," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 795–816.

[18] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complexity*, vol. 2020, 2020.

[19] O. Kamal, A. Kumar, and T. Vaidhya, "Hostility detection in hindi leveraging pre-trained language models," in *International Workshop on Combating On line Ho st ile Posts in Regional Languages dur ing Emerge ncy Si tuation*. Springer, 2021, pp. 213–223.

[20] R. Joshi, P. Goel, and R. Joshi, "Deep learning for hindi text classification: A comparison," in *International Conference on Intelligent Human Computer Interaction*. Springer, 2019, pp. 94–101.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[22] A. Baruah, K. A. Das, F. A. Barbhuiya, and K. Dey, "Automatic detection of fake news spreaders using bert." in *CLEF (Working Notes)*, 2020.

[23] A. M. Babu, J. Catharine, K. Divya, and A. Joffi, "Hawkseye-a machine learning-based technique for fake news detection with iot."

[24] M. Aldwairi and A. Alwahedi, "Detecting fake news in social media networks," *Procedia Computer Science*, vol. 141, pp. 215–222, 2018.

[25] M. U. Islam, M. M. Hossain, and M. A. Kashem, "Covfake: A word embedding coupled with lstm approach for covid related fake news detection," *International Journal of Computer Applications*, vol. 975, p. 8887.

[26] M. Bhardwaj, M. S. Akhtar, A. Ekbal, A. Das, and T. Chakraborty, "Hostility detection dataset in hindi," *arXiv preprint arXiv:2011.03588*, 2020.