Detection Of Fake News Using Deep Learning Based Models

Likitha Santhapalli[#1], Divya Reddy Sadhula[*2]

Department of Computer Science

University of Central Florida
Orlando, Florida, United States

[1] li466220@ucf.edu

[2]di068528@ucf.edu

***Abstract*— In today's digital world, people prefer to engage through various online platforms such as Twitter, Facebook, WhatsApp, etc. These applications not only serve as a medium for communication but also as a source of significant information. However, these online platforms' popularity and ease of use have also made it easier for false news to spread, which has resulted in misinformation. This fake news may impact in various ways such as social unrest, dividing public opinion, the health sector, political elections, stock markets, etc. As Artificial intelligence evolves detection of this news is a challenging task. This paper focuses on the implementation of deep learning models such as LSTM, BiLSTM, and the hybrid approach CNN-BiLSTM to detect fake news. By combining the best features of recurrent neural networks (RNNs), like LSTMs, and convolutional neural networks (CNNs), these models efficiently process and analyze textual data. While the LSTM and BiLSTM components are skilled at capturing temporal dependencies in the data, the CNN components are proficient at extracting high-level feature representations from the text. We have used the Information Security and Object Technology (ISOT) dataset which contains thousands of fake and truthful articles collected from legitimate sources which helps us evaluate each model's performance.**

Keywords—Fake news, deep learning, detection of fake news, LSTM, Bi-LSTM

1. INTRODUCTION

Fake news is pervasive nowadays and is too easy to spread on social media. With the rapid growth of technology, the use of social media platforms has significantly increased which resulted in the spread of false information quickly. There is no doubt that social media has brought a lot of convenience to people. However, in the millions of information available in the internet and being shared daily, people find it difficult to differentiate the credibility of the news whether it is fake or real news.

It can be spread in various forms such as satirical or parody content, manipulated content, and completely fabricated stories. Consequently, it may harm people who mistakenly believe fake news is real, which may lead to the loss of financial and personal information.

One such instance is fake news that misguided many people during the COVID-19 pandemic regarding vaccines impacting the health sector and also taking the lives of 800 people globally and also spreading false information about vaccines. Apart from this, stock markets have also been impacted because of false news causing panic among investors and selling off their holdings at once. Given this, fake news detection is required to help people differentiate the credibility of the news so that readers would not be affected by the fake news.

In this paper, we propose a hybrid deep-learning approach consisting of CNN and LSTM layers. We have used the models including LSTM, BiLSTM, and CNN-BiLSTM. The CNN-BiLSTM layer comprises of embedding layer, convolution layer, max-pooling layers, and dense layers. The two regularization techniques have been used to solve the overfitting problem. We have evaluated the performance of each model and experimental results show that the proposed models have shown a better accuracy and performance metrics and are less prone to overfitting.

This paper proposes a brief explanation of these models and performance metrics. After this introduction, Section 2 discusses the related works, Section 3 describes our dataset, including data preprocessing, and Section 4, explains the classification of models. Section 5 discusses Results and analysis, and Section 6 concludes with an overview of our research and suggests directions for future work.

2. RELATED WORKS

Machine learning and deep learning methods have shown great results in the detection of fake news. In this section, we will review some previous works.

In this paper [1], Pushkar, etc., have used deep learning models such as CNN, and LSTM to identify fake news detection and achieved the highest accuracy of 83% for both models. In [2], the author has reviewed numerous methods to predict false news using natural language processing techniques that are used to process the data and prepare it for the use of deep learning and machine learning algorithms.

In [3], the three methods include ELMO sentence representations by combining it with CNN, TD-IDF for feature extraction with SVM classifier, and TD-IDF with logistic regression have been used by authors in this study to verify the truthiness of the content present in Twitter in English and Spanish. They have used a PAN2020 profiling corpus dataset. As a result, for English text, they have achieved an accuracy of 65% using CNN and 70% using SVM. For Spanish, the accuracy is 80% using SVM and 78% using Logistic regression.

An ensemble classification model for the detection of fake news was proposed by Saqib et al. [4] and has proven to be more accurate than the state-of-the-art when tested on the ISOT fake news dataset. Mohammed et al. [5], applied various n-gram levels to extract features from the ISOT false news dataset. Sebastian, et al. [6] also performed text analysis and false news detection on the ISOT fake news dataset using a neural network-based method.

In this study [7], they applied the BiLSTM model to the ISOT dataset, and as the outcome, they achieved an accuracy of 99.82%. Authors of [8], employed Random Forest, Logistic Regression, and Passive Aggressive techniques, using datasets from GitHub and Kaggle. While Random Forest and Logistic Regression have achieved an accuracy of 79% and 81% respectively, the passive-aggressive model has reached a maximum accuracy of 92%.

The buzz feed dataset, which contains news from Facebook posts, was used to test the KNN classifier in this study [9]. The accuracy result was 79%. To identify fake news, Khan et al. [10] evaluated the efficacy of deep neural network techniques versus traditional machine learning techniques. In addition, stop word removal, stemming, IP and URL removal, and data preprocessing were carried out. Lexical and emotional features for n-grams were extracted for every model

3. DATA AND METHODOLOGY

3.1 Dataset

In this paper, the ISOT Fake news dataset has been used which contains thousands of articles and news collected from various legitimate and truthful sources by politifact.com. The dataset has two CSV files: one with True news and the other with Fake news.

To gain insight into the fake and real news, created a word cloud respectively. The word cloud of the true news dataset is displayed in Figure 1. And the word cloud for the fake news dataset is displayed in Figure 2.

From these two figures, we can observe that the words "Trump," "Russia," "say," and "House" are commonly found in real news articles, while "VIDEO," "Trump," "Watch," and "Hillary" are frequently mentioned in fake news articles. We can also observe it has unnecessary words which does not help us in the detection of fake news. The text has to be cleaned to train the models on proper data. We have combined both datasets and split them into training, testing, and validation datasets at the ratio of 80%:10%.:10%

Fig.1. Word cloud for True news dataset

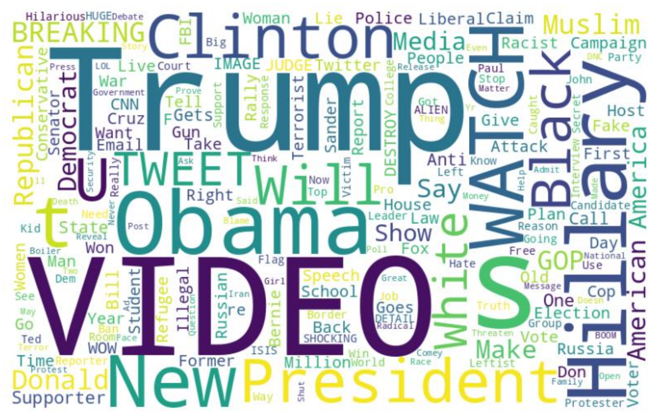

Fig.2. Word Cloud for fake news dataset

3.2 Data Preprocessing

The main aim of this project is to use NLP techniques for processing the input data. The dataset contains test, title, subject and date columns. We have trained the model only on the title data. Each title is about 42 words long, while the text is approximately 8135 words long. A preprocessing pipeline has been built for each statement to eliminate the noise in the dataset. To establish a preprocessing pipeline using the Natural Language Toolkit (NLTK), the necessary libraries and functions have been imported.

We have removed the unnecessary columns such as subject and date. All the text has been converted to lowercase using the lower function. The subfunction of Regular Expression (re) has been used to replace special patterns in the text such as punctuations, square brackets, numbers, and links.

Using the NLTK library, the "stopwords" and "punkt" have been downloaded, and the process of removing stopwords has been implemented. These stopwords do not add any weight to our data and also increase the unnecessary size of the dataset. After that, the word tokenizer function has been used for breaking the text and cleaning the dataset. Additionally, stemming has been applied to each word, reducing them to their root forms which helps in improving the overall accuracy of the prediction outcome by removing unnecessary noise from the dataset

3.3 Feature Extraction

It is important to convert the dataset into a form that models can handle. In our study, we prepared the text data for analysis for deep learning models using Tokenization and Glove embeddings.

Tokenization is a method of transforming words into corresponding numerical values which is used to implement the model. In this method, each word is transformed into a sequence of integers, where each integer represents a unique token. After tokenization, padding is used to make the length of data uniform, which is crucial because deep learning models require input data of consistent size. Hence, it is concluded that padding and tokenization as important steps in building deep learning models.

Additionally, we enhanced our data with Glove embeddings, which are pre-trained vectors that represent words based on their meanings and common usage. Glove or Global Vector for Word Representation is a word-inserting method that uses unsupervised learning that concatenates the global words and word occurrence matrix to generate the embedding. This method is applied using the preprocessing library TensorFlow's Kera's to define the relationship and differentiate between the words based on their vector representation.

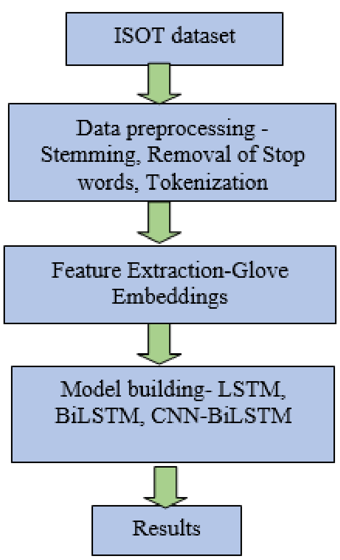

Fig.3. Flow of the Framework

4. MODELS

The models that have been implemented in this paper are based on deep learning neural network models. The experimentation is performed on LSTM, BiLSTM, and hybrid CNN-BiLSTM. The detailed explanation of the classifiers is as follows:

4.1 RNNs

A recurrent neural network (RNN) is a type of deep learning network model designed to convert and process sequential data input such as words, sentences into a specific sequential data output. An RNNs mimics human such as translating text from one language to another, though its network of interconnected layers.

4.2 LSTM

LSTMs, or Long Short-Term Memory networks, are a particular type of RNN, or Recurrent Neural Network are designed to process sequential data and LSTMs are an advanced version that addresses some of the limitations of traditional RNNs.

LSTMS are unidirectional. The LSTM consists of three layers or gates: the input gate, the output gate, and the forget gate. The forget gates identify the information that is forgettable and unimportant based on the input of the LSTM layer. The input gates decide which data is important and stored in memory. On the other hand, the output gates decide which memory cell value should be the output.

Sequential data, including time series, text, and speech, can be processed and analyzed by LSTMs. To avoid the vanishing gradient issue that affects conventional RNNs, they use a memory cell and gates to control the flow of information. This allows them to select the information as needed or discard them. Our LSTM model consists of an embedding layer, LSTM, and two dense and dropout layers as shown in Fig.7.

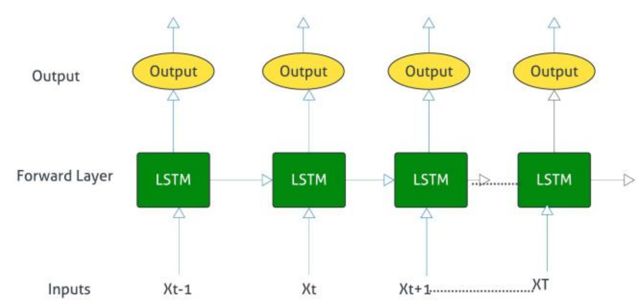

Fig.4. Architecture of LSTM model

4.3 BiLSTM

Bidirectional LSTM or BiLSTM sequence model contains two LSTM layers, one for processing input in the forward direction and the other for processing in the backward direction. The idea behind this approach is that by processing data in both directions, the model can better understand the relationship between sequences. BiLSTM model is more suitable for tasks like sentiment analysis, text classification and machine translation. The architecture of BiLSTM comprises of two unidirectional LSTMs which process the sequence in both forward and backward directions.

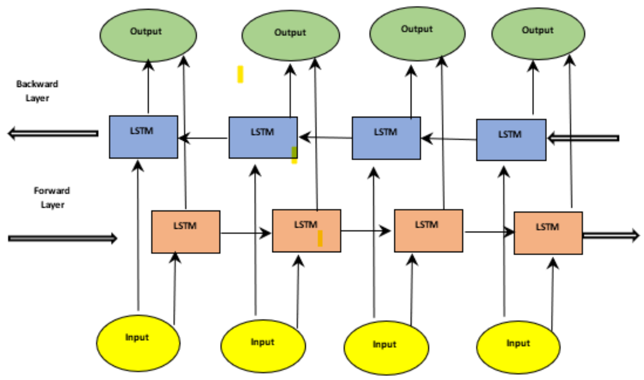

Fig.5. Architecture of BiLSTM Model

4.4 CNN- BiLSTM

We have 1D CNN layer before BiLSTM layer. The output of the CNN layer is then fed to BiLSTM layer. By combining CNN and BiLSTM layers allows the model to capture both short range patterns and long-term dependencies within the input sequences. This is a hybrid model that combines features of both RNNS and CNNs.

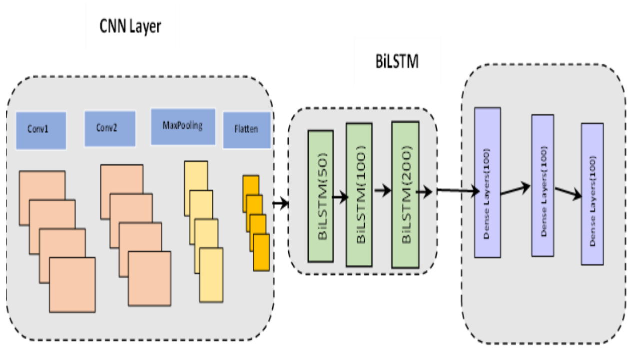

Fig.6. Architecture of CNN-BiLSTM Model

4.5 Calculations of Predictions

The precision, recall, accuracy, and F1-score are a few of the evaluation measures that mostly determine the outcomes of machine learning models. Precision expresses the percentage of true positive predictions among all positive predictions produced, hence showing the quality of the model's projected values. On the other hand, recall measures the number of relevant predictions the model made, expressing the true positive predictions as a percentage of the real positive values.

The harmonic mean of precision and recall is the F1-score, also referred to as the F-measure or F-beta score. Another important evaluation metric is accuracy which measures the proportion of correct predictions made by the model out of the total number of predictions. The formula for calculating F1-score, precision, recall and accuracy is given in Equation 1.

$$\text{Precision} = \frac{\text{True Positives}}{(\text{True Positives} + \text{False Positives})} \quad (1)$$

$$\text{Recall} = \frac{\text{True Positives}}{(\text{True Positives} + \text{False Negatives})} \quad (2)$$

$$\text{F- Measure} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})} \quad (3)$$

$$\text{Accuracy} = \frac{\text{Total number of correct predictions}}{\text{Total number of predictions}} \quad (4)$$

Equation.1. Formula for Precision, Recall, F-Measure and accuracy

5. RESULTS AND ANALYSIS

The performance of the classification models, including LSTM, BiLSTM, and CNN BiLSTM, was evaluated on the dataset using various evaluation metrics. The experiments were conducted on Google Collaboratory, leveraging its open-source environment for Python. The training, testing and validation datasets are 80%,10% and 10% respectively. In compilation of classifier, the binary cross-entropy is used as loss function along and the adam optimizer along with metrics as accuracy. The activation function used is Sigmoid and ReLU. We have used a Glove embedding of 100 dimensions and the value of epoch is 10. The summary of the classifiers LSTM, BiLSTM, and CNN BiLSTM are given in Fig 7 to Fig 16.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 31, 100)           1195000

 dropout_2 (Dropout)         (None, 31, 100)           0

 lstm_1 (LSTM)               (None, 128)               117248

 dropout_3 (Dropout)         (None, 128)               0

 dense_2 (Dense)             (None, 64)                8256

 dense_3 (Dense)             (None, 1)                 65

=================================================================
Total params: 1320569 (5.04 MB)
Trainable params: 125569 (490.50 KB)
Non-trainable params: 1195000 (4.56 MB)
_________________________________________________________________
```

Fig.7. Summary of LSTM model

	precision	recall	f1-score	support
0	0.96	0.92	0.94	3552
1	0.91	0.95	0.93	3183
accuracy			0.94	6735
macro avg	0.93	0.94	0.94	6735
weighted avg	0.94	0.94	0.94	6735

Fig.8. Classification Report of LSTM

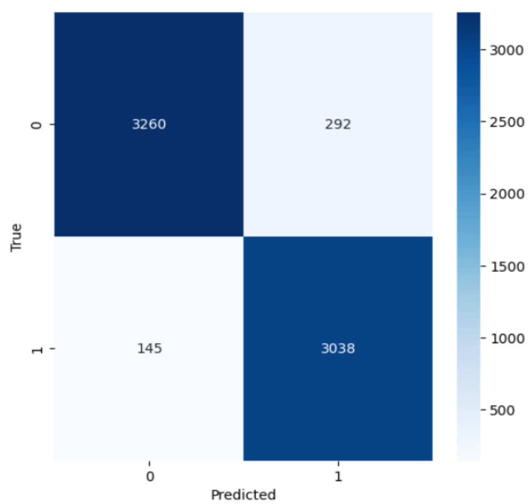

Fig.9. Confusion Matrix of LSTM

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 31, 100)           1195000

 dropout_4 (Dropout)         (None, 31, 100)           0

 bidirectional (Bidirection  (None, 256)               234496
 al)

 dropout_5 (Dropout)         (None, 256)               0

 dense_4 (Dense)             (None, 64)                16448

 dense_5 (Dense)             (None, 1)                 65

=================================================================
Total params: 1446009 (5.52 MB)
Trainable params: 251009 (980.50 KB)
Non-trainable params: 1195000 (4.56 MB)
_________________________________________________________________
```

Fig.10. Summary of BiLSTM

Classification Report:

	precision	recall	f1-score	support
0	0.96	0.93	0.94	3552
1	0.92	0.95	0.94	3183
accuracy			0.94	6735
macro avg	0.94	0.94	0.94	6735
weighted avg	0.94	0.94	0.94	6735

Fig. 11. Classification Report of BiLSTM

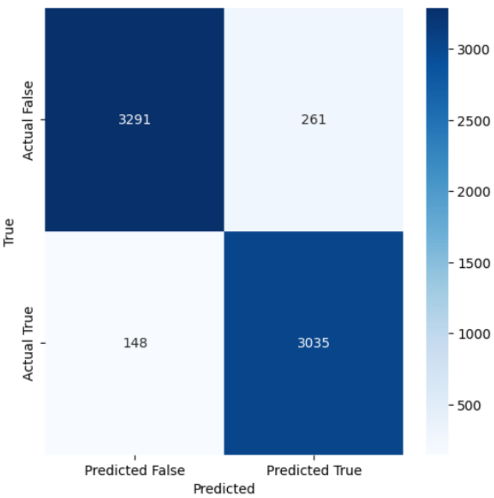

Fig. 12. Confusion Matrix of BiLSTM

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 31, 100)           1195000

 dropout_6 (Dropout)         (None, 31, 100)           0

 conv1d (Conv1D)             (None, 31, 64)            19264

 max_pooling1d (MaxPooling1  (None, 15, 64)            0
 D)

 bidirectional_1 (Bidirecti  (None, 256)               197632
 onal)

 dropout_7 (Dropout)         (None, 256)               0

 dense_6 (Dense)             (None, 64)                16448

 dense_7 (Dense)             (None, 1)                 65

=================================================================
Total params: 1428409 (5.45 MB)
Trainable params: 233409 (911.75 KB)
Non-trainable params: 1195000 (4.56 MB)
_________________________________________________________________
```

Fig.13. Summary of CNN BiLSTM

Classification Report:

	precision	recall	f1-score	support
Class 0	0.96	0.89	0.92	3552
Class 1	0.89	0.95	0.92	3183
accuracy			0.92	6735
macro avg	0.92	0.92	0.92	6735
weighted avg	0.92	0.92	0.92	6735

Fig. 14. Classification Report of CNN BiLSTM

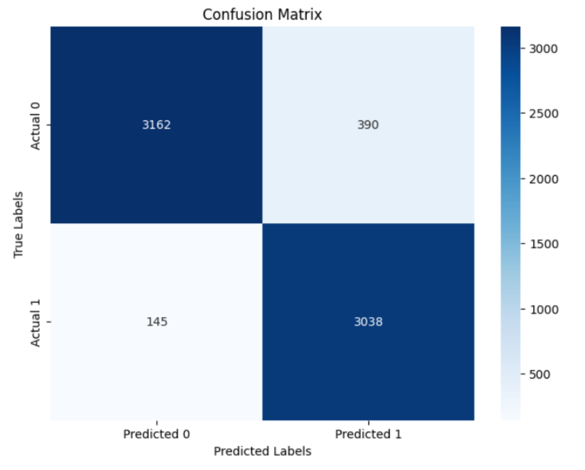

Fig. 15. Confusion Matrix of CNN BiLSTM

The results demonstrate that all models achieved relatively high accuracies, with BiLSTM performing slightly better than LSTM and CNN-BiLSTM. For both classes, BiLSTM gave the highest overall precision and F1 scores. Still, it is worth mentioning that LSTM and BiLSTM performed almost identically but for slight differences in metrics. On the other hand, among the three models, the CNN BiLSTM model had a lower precision for Class 1 and slightly lower overall accuracy. In summary, though all of them did perform well as indicated by their results, based on accuracy and other performance measures such as mentioned above only subtle differences exist between them; however, it may be noted that generally speaking BiLSTM, outperformed LSTM and CNN-BiLSTM in terms of its accuracies as well as other performance indicators.

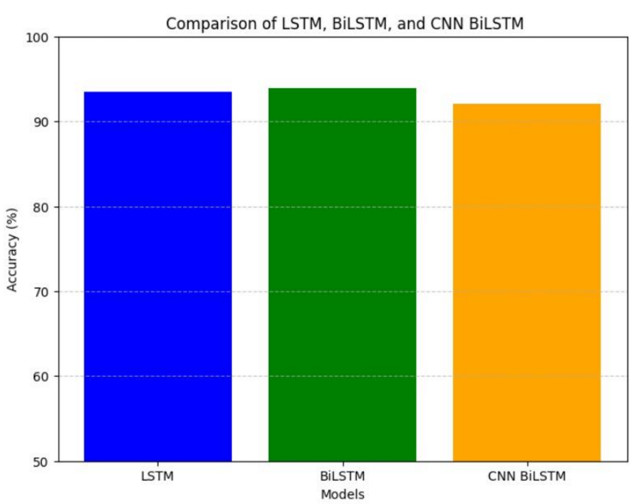

Fig. 16. Accuracy graphs for DL classifiers

From Fig 16, we can observe the accuracy for LSTM, BiLSTM and CNN-BiLSTM are 93.51%, 93.93%,92.06% respectively.

6. CONCLUSION AND FUTURE WORK

As technology continues to advance and the use of social media becomes more widespread, there are negative consequences, including the extensive spread of fake news. In this paper, we have implemented deep learning models on ISOT dataset to detect the fake news. The glove embeddings are used in the feature extraction to get global co-occurrence of words vectors as it doesn't depend on the local data. The highest accuracy resulted in Bi-LSTM model compared to LSTM and CNN-BILSTM model with accuracy of 93.93%.

Moreover, future investigation areas include hyperparameter tuning techniques for further optimization of these classifiers. Therefore, if we systematically change the model parameters, better accuracy or predictive performance can be achieved. Finally, these improvement processes help improve the strength and efficiency of deep learning models for different applications enhancing natural language processing as well as text classification field.

REFERENCES

[1] Pushkar and Rakesh., "Fake News Detection using Deep Learning Based Approach", International Conference on Circuit Power and Computing Technologies (ICCPCT),2023

[2] P. Sharma and R Sahu, "Review of Machine Learning and Deep Learning based Methods for Phoney News Detection",10th International Conference on Computing for Sustainable Global Development (INDIACom), 2023.

[3] S. Ahmed et. al, "Development of fake news model using machine learning through natural language processing", preprint arXiv:2201.07489, 2022.

[4] Hakak, S., Alazab, M., Khan, S., Gadekallu, T. R., Maddikunta, P. K. R., & Khan, W. Z., An ensemble machine learning approach through effective feature extraction to classify fake news. Future Generation Computer Systems, 117, 47-58,2021

[5] Goldani, M. H., Momtazi, S., & Safa bakhsh, R.," Detecting fake news with capsule neural networks", Applied Soft Computing, 101, 106991,2021

[6] Kula, S., Choraś, M., Kozik, R., Ksieniewicz, P., &Woźniak, M., "Sentiment analysis for fake news detection by means of neural networks". In International Conference on Computational Science (pp)653-666). Springer, Cham., June,2020

[7] T. Jiang et. al, "Fake News Detection using Deep Recurrent Neural Networks, "in 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2020.

[8] P. Faustini et. al, "Fake news detection using one-class classification," in 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), 2019.

[9] B. Al Asaad et. al, "A tool for fake news detection," in 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2018.

[10] Khan, J. Y., Khond Aker, M. T. I., Iqbal, A., & Afroz, S. (2019). A Benchmark Study on Machine Learning Methods for Fake News Detection. 1–14. http://arxiv.org/abs/1905.04749

[11] Ahmed, H., Traore, I., & Saad, S. (2018). Detecting opinion spams and fake news using text classification. Security and Privacy, 1(1), e9.

[12] Lazer, D. M., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., ... & Zittrain, J. L. (2018). The science of fake news. Science, 359(6380), 1094-1096.

[13] Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter, 19(1),22-36.

[14] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey,

W., ... & Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

[15] A. Goldman C. Kang. In washington pizzeria attack. Fake News Brought Real Guns, 5:102437,2016

[16] [13] Yimin Chen, Victoria L Rubin, and Niall Conroy. Towards news verification: deception detection methods for news discourse. 2015.