International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 05 | May 2021 www.irjet.net p-ISSN: 2395-0072

Single Modal and Bimodal Approach to Fake News Detection

Jash Vyas[1], Shubham Bhalerao[1], Meet Gogri[1] , Kartiki Sanap[1], Prof.Seema Shrawne[2]

[1]B.Tech Student, Dept of Computer Engineering & IT , VJTI College, Mumbai, Maharashtra, India.
[2]Assistant Professor, Dept of Computer Engineering & IT, VJTI College, Mumbai, Maharashtra, India.

---***---

Abstract - *With the expeditious digitization across all channels and mediums, data and information are now considered to be the greatest asset in the world. However, many times the authenticity of information on the internet is compromised which has resulted in the problem of fake news. In recent times, the menace of fake news has been amplified by the exponential growth of social media with fake news spreading at such a rapid rate that it has the potential of causing real world impact to millions of users within minutes. Thus, an automated approach to detecting fake news has become a need of the hour. As an effort in this direction, this paper focuses on automating the task of fake news detection using machine learning. In this paper we explore the application of natural language processing and deep learning techniques to identify fake news using a single modal (text based) and bimodal (text and image based) approach. Several pre-processing techniques such as stemming, tokenization, stop words removal, glove embeddings,etc are used to convert raw data into suitable form. Then a fake news detection model is trained on the processed data. We start by exploring basic classifiers such as logistic regression, k nearest neighbours, etc and then move on to deep learning models such as CNNs and LSTMs. We achieve best results using a hybrid CNN-LSTM model which combines the capabilities of CNNs and LSTMs.*

Key Words: Fake News Detection, Deep Learning, Natural Language Processing, Convolutional Neural Networks, Long Short Term Memory, Glove Embeddings

1. INTRODUCTION

In today's technological world, a huge amount of data gets generated online every day. However most of the data which is flooded on the internet is fake and it is generated to attract the audience, to misguide people, to influence beliefs and decisions of people. Obtaining and spreading information through social media platforms has become extremely easy, which makes it difficult and nontrivial to detect fake news based merely on the content of news.

Several tech companies like Google,Facebook.Twitter have attempted to address this particular concern of fake news spread. Majority of the countries all across the world are trying to combat this challenge of fake news spread. Users continue to deal with the sites containing false information and whose involvement tends to affect the reader's ability to engage with actual news. Most of the sites which contain such information also include a sharing option that implores users to disseminate the contents of the web page further. Social networking sites allow for efficient and fast sharing of material and thus, users can share the misleading information within a short time. Thus, to identify fake news and impede the spread of fake news is of pivotal importance in today's age.

2. LITERATURE REVIEW

Fake news detection is a very well-established and well-researched task in NLP. It is defined as determining from the news article including the head, body, images,etc whether the news can be determined as real or fake. Previously many researches have been conducted to detect fake-news on the internet and social media platforms using NLP tools and machine learning techniques.

The Fake News Challenge Stage-1 (FNC-1) which was held in 2017 featured many novel solutions for this problem. Riedel et al. and other teams who won in FNC-1 achieved accuracy close to 82% in the stance detection stage. After the completion of this competition, many different solutions have emerged using various NLP Techniques. The datasets used in FNC-1 are available publicly and we are close to having standard benchmarks to compare all the newly proposed techniques. Currently there are many other datasets available for fake-news detection which have been mentioned in [10],[11].

Monther Aldwairi et al. [1] proposed a solution to detect and filter out sites containing false and misleading information on the internet, using important features of the title and post to accurately identify fake posts.Different features like titles starting with numbers, all caps words, presence of question and exclamation marks, whether the user left the page immediately, and content related to the title- were extracted from the content of the webpage whose URL is provided. WEKA classifiers like Random Forests, Naive Bayes, Bayes network, Logistic Regression etc. were used to train the model. Most of the websites considered in this paper were from social media like Facebook, Forex and Reddit.

M. Umer et al. [2] has proposed a hybrid Neural Network architecture that combines the capabilities of CNN and LSTM for the detection of fake-news. The network architecture is used with two different dimensionality reduction approaches, Principal Component Analysis (PCA) and Chi-Square. The dataset used for training this hybrid model was the FNC-1 dataset. Also a comparison was made between the metrics of the proposed solution and already existing solutions like TalosComb, TalosTree, stackLSTM, BERT, XLNet, etc. Some researchers used different variations of LSTMs like BiDirectional LSTMs[18], LSTM attention models[9], etc.

Julio C et al. [3] has proposed supervised machine learning techniques to classify fake news. This paper roughly classifies its features into three types : 1) Features from news content, 2) Features from news source and 3) features from the environment. These features are then fed into various classifiers like random forest, XGBoost, Naive Bayes and Support Vector Classifier to classify fake news.

In [11], Fakeddit presents a multimodal approach to detecting fake news . It has a multimodal dataset which consists of more than 1 million samples through multiple categories of fake news. They processed the dataset through several stages of review and labelled all of the samples for a 2-way, 3-way or 6-way classification by following a distant supervision. They constructed hybrid models consisting of both text and images to perform various extensive experiments so as to form several different variations of classifications which in turn shows the importance for multimodality and fine-grained classifications which is the novel aspect absolutely unique to Fakedd .

For multimodal fake-news detection, numerous deep learning architectures have been proposed including MVAE(Multimodal Variational Autoencoder for Fake News Detection)[15], Hybrid LSTM-CNN architectures, EANN(Event-Adversarial Neural Networks)[12], etc. A bimodal variational autoencoder coupled with a binary classifier is used in MVAE for detecting fake news.The variational autoencoder is capable of learning probabilistic latent variable models by optimizing a bound on the marginal likelihood of the observed data. The multimodal representations obtained from the bimodal variational autoencoder are utilized by the fake news detector to classify posts as fake or not.

EANN is an end-to-end framework which can derive event-invariant features and thus benefit the detection of fake news on newly arrived events. In EANN, extracting the textual and visual features from posts is performed by the multi-modal feature extractor. It cooperates with the fake news detector to learn the discriminable representation for the detection of fake news. For the removal of event-specific features and to keep shared features among events, EANN uses the event discriminator.

3. DATASET DESCRIPTION

We gathered various datasets in specific as well as in general categories for analysis of Fake & Real News.

For the Single Modal (Text based) approach to fake news detection, we considered the ISOT fake News dataset [19], [20] consisting of 44898 world news articles from 2016 to 2017.

News	Size (Number of articles)	*Subjects*	
Real-News	21417	**Type**	**Articles size**
		World-News	*10145*
		Politics-News	*11272*
Fake-News	23481	**Type**	**Articles size**
		Government-News	*1570*
		Middle-east	*778*
		US News	*783*
		left-news	*4459*
		politics	*6841*
		News	*9050*

Fig -1: ISOT Fake News Dataset

The dataset has attributes like title of the article, text (body) of the article, subject of the article (political news), publication date and output variable - label (fake or real).

For the Bimodal (text and image based) approach to fake news detection, we considered a dataset obtained from fakeedit paper comprising 7336 news articles[11].

The dataset has attributes like title of the article (clean_title), domain, hasImage(True or False indicating if article has image), id(identifier), Image_url(image associated with article), linked_submission_id, num_comments(number of comments on posts), score, subreddit of post, title of post, upvote_ratio and output variable -2_way_label (indicating if news is real or fake).

4. DATA PRE-PROCESSING

Several tasks for text and image pre-processing were performed on the training dataset before it can be used to train our model. In this section we describe various NLP and few other image pre-processing techniques that we have used.

Text Preprocessing:
For text-based classification, we have the title and body of the news article. So we concatenated the two into a single string before preprocessing it.

Lowercase: We first convert the text that is the headline and the content of the news to lowercase as case sensitivity doesn't make any difference in the news.

Remove Punctuation: As punctuation does not enhance our understanding of news text, we have removed them as well.

Remove Whitespaces: The above steps can result in extra whitespaces being added to our text. As these extra whitespaces are of no use they are replaced by a single space.

Remove Stopwords: Stopwords are very common words that exist in the text and have very minor importance in terms of features e,g 'of', 'the', 'and', 'an', etc. Hence we removed the stopwords which helped in reducing the processing time and saving memory.

Stemming and Lemmatization: To reduce our vocabulary and dimensionality for NLP tasks and to improve speed and efficiency , we can use stemming and lemmatization. Using these techniques the words are reduced to simpler forms. The difference between the two is that lemmatization converts the word to the root word or base word whereas stemming just converts it to a simpler form called stem. As compared to lemmatization, stemming is a simpler, faster process and for simpler use cases, it can have the same effect as lemmatization. and hence we have used only stemming in our solution.

For training purposes, the textual data must be converted into numeric data before feeding it to the model. Hence we use different preprocessing techniques used in NLP like stemming, padding, vectorization, tokenization, one-hot encoding, glove embeddings, etc.

One-hot encoding: In one hot encoding, every word (even symbols) of the text is represented using a vector of only 1s and 0s. So if we have a vocabulary of 50,000 different words each word will be represented by a vector of size 50,000 consisting of 1 and 0. Since every vector only has a single 1 and rest of them are 0s so it wastes a lot of space. So we decided not to go with this because our word vocabulary consisted of almost 200000 different words and hence using one-hot encoding would increase computing time drastically.

Tokenization: It is a method of separating text into smaller units called tokens. Every word will be a unique numeric value using tokenization. Now each sentence can be represented as a list of numbers. A vocabulary can be maintained to keep the list of unique words present in the dataset which can help to assign the same value every time the words appear.

Word vectorization: It is a methodology which maps words or phrases from vocabulary to a corresponding vector of real numbers which are used to find word predictions,word similarities/semantics. Different Vectorizers like count vectorizer and tfidfvectorizer are available in sklearn which can be used to convert words directly into a vector of fixed size.

Another method which can be used for representing text as a vector is Word Embedding. There are different word embedding techniques like Word2Vec , CBOW(Continuous Bowl of Words), Glove Embeddings, etc. An embedding layer can be added in the learning model which can be trained to find optimal mapping of each of the unique words to a vector of real numbers. Glove Embeddings are pretrained embeddings using the Wikipedia and GigaWord dataset. We have tried both methods - training our own embeddings and the pre-trained Glove embeddings. We convert every word to a 100 dimensional vector using these word embeddings.

Padding: After converting the textual data into numeric form , all sentences of different length are converted to equal length data using padding. We have zero-padded the inputs smaller than 700 words.

Data Cleaning : We got rid of the fields in the dataset that would not be useful in the detection process like date, subject,ids, etc.We also got rid of several columns for the domain validation and impact scoring.

Image Preprocessing:
Image preprocessing task involves conversion of image data to numeric data.

Converting to RGB Format: For converting image data to numeric data, input images are initially converted to RGB to get all images in the same colour format.

Converting to Numerical Form: Further processing consists of converting the RGB images to a numpy array to have a numerical form of the image.

Resizing Images: After completing the above steps we stack them up in a final array which can then be used for reshaping and matching the shape of the textual format and then concatenating them for a final input array.

5. MODEL DESCRIPTION

After successfully applying the preprocessing steps our dataset is ready for training. We apply different models to train our dataset and compare them to find the model which best fits our data.

Single Modal:
For single modal, we first started by applying some basic and standard classifiers such as logistic regression, k nearest neighbours, naive bayes and random forest classifier.

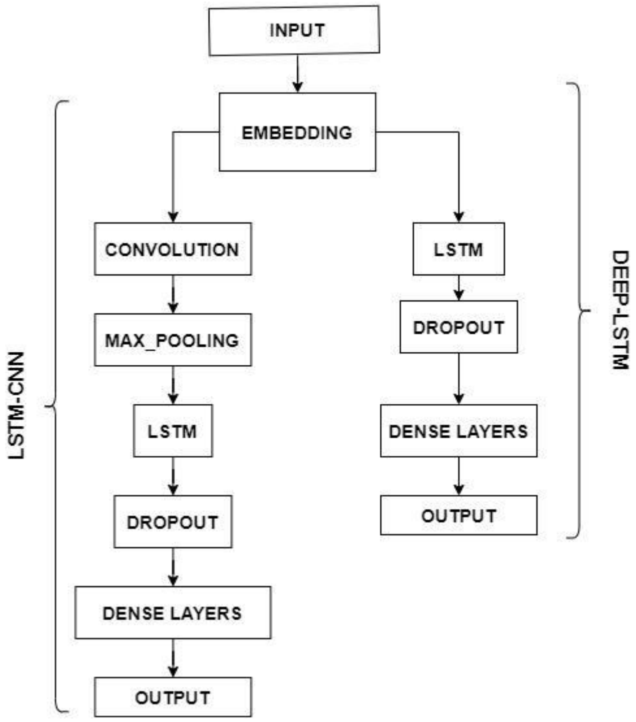

Fig -2: Architecture of two single modal models

After applying these basic models, we applied deep learning models such as feed forward neural networks, CNNs and LSTMs. Since we are dealing with sequence data, we applied a deep LSTM model to our dataset. LSTM model helps in finding out longer range dependencies in data. It also solves the problem of vanishing gradients. After this model, we applied a CNN- LSTM Model on our data. The intuition behind this model is that it combines the power of both CNN and LSTM Models and gives us better results than if these models were used in isolation. CNN is useful in extracting features from the textual data and also helps reduce the number of parameters saving both memory and processing time.

We have used these models with two approaches: 1) with glove embeddings and 2) without glove embeddings. We got better results when we used our model with glove embeddings.

Here we describe the LSTM CNN Model with glove embeddings which gave us the best results:
Our first layer is an embedding layer which takes the input headlines and the body of the article and converts every word into a vector of size 100 . These vectors are passed to CNN layers for extraction of contextual features. Then the output of the CNN layers is passed on to the LSTM layer after which is passed on to a series of fully connected dense layers having sigmoid function at last which produces the final prediction. Dropout layers are added to prevent our model from overfitting. We have used binary cross entropy as our loss function and adam optimizer with a batch size of 64 for training.

BiModal:

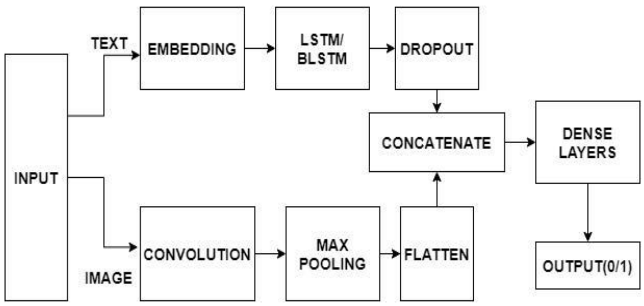

Fig -3: Architecture of Bi-modal model

For BiModal detection, we again have used multiple models and the LSTM-CNN model gave us the best results. Our first layer is an input layer which takes the input of image data. The input matrix with shape of samples*480*960 is then passed on to the CNN layers. After the CNN layers, another input layer is used for the textual part of the data and then passed on to an embedding layer which converts every word to a vector of size 100. It is then passed to a LSTM layer. In contrast with single modal, this time before the dense layers, there is a concatenate layer which combines the output of both the CNN layers(Images) and the LSTM layers (Text) .

After the concatenating layer, the output is then passed to a series of fully connected dense layers having a sigmoid function at last which produces the final prediction. Similar to the single modal model, dropout layers are added to reduce overfitting and the training is carried out with a batch size of 64 using adam optimizer and binary cross entropy as a loss function.

6. RESULTS

After training our model it's now time to test the efficacy of our models. We performed a randomized train test split on our dataset in a 70/30 (85/15 for bi-modal) ratio meaning 70% of the dataset is used for training the model and the rest 30% is used for testing the model. The models are evaluated based on various metrics used for classification problems like accuracy, precision, recall and f1 score.

6.1 Statistics

Single Modal:

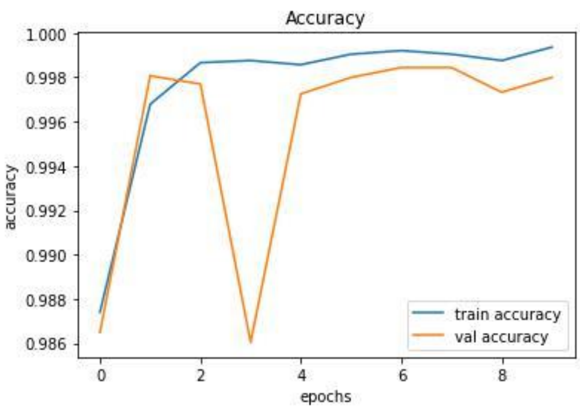

Fig -4: LSTM CNN with Glove Embeddings

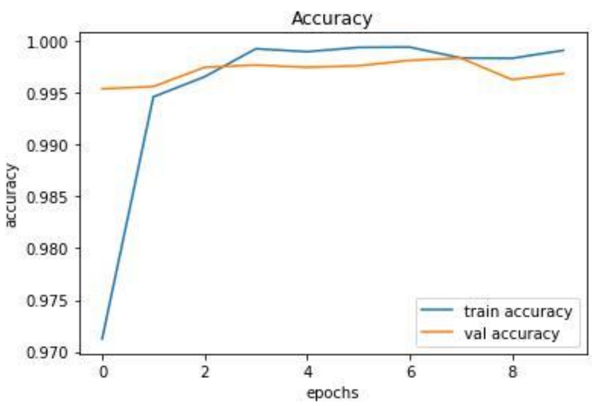

Fig -5: Deep LSTM With Glove Embeddings

Bi-Modal:

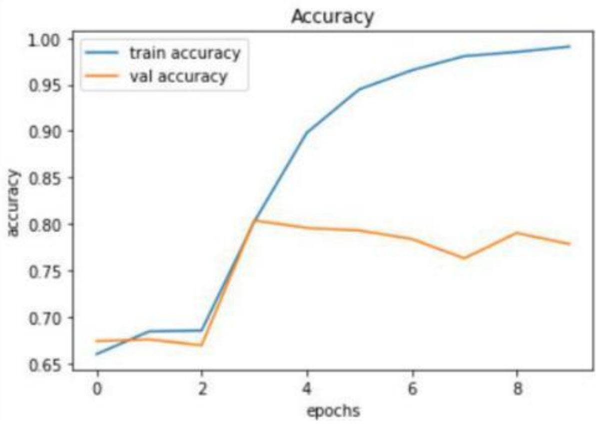

Fig -6: LSTM CNN Model

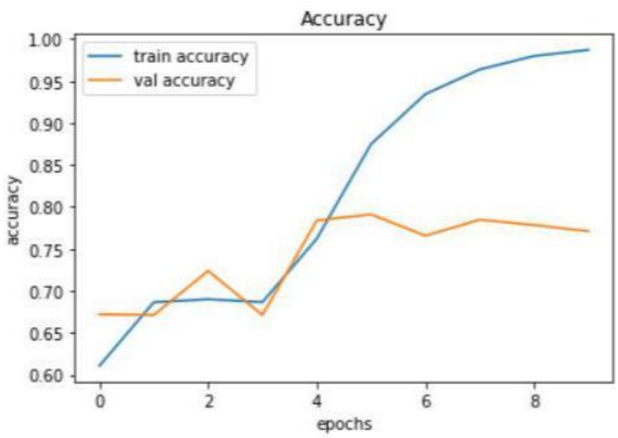

Fig -7: Bi-LSTM-CNN

6.2 Performance Comparison

Single Modal :

Classification Models	Accuracy	Precision	Recall	F1-Score
LSTM-CNN (with Glove embeddings)	0.9984	0.9981	0.9984	0.9982
Deep-LSTM (with Glove embeddings)	0.9969	0.9981	0.9984	0.9967
LSTM-CNN (without Glove embeddings)	0.9978	0.9967	0.9986	0.9977

Deep-LSTM (without Glove embeddings)	0.9967	0.9961	0.9969	0.9965
Naive Bayes (With Count Vectorizer)	0.9530	0.9489	0.9518	0.9503
Naive Bayes (With TfidfVectorizer)	0.9366	0.9382	0.9269	0.9324
Random Forest	0.9893	0.9873	0.9902	0.9887
K-Nearest Neighbour	0.8245	0.8231	0.8004	0.8115
Logistic Regression (with pipeline module)	0.9826	0.9826	0.9805	0.9815
Logistic Regression (without pipeline)	0.9941	0.9948	0.9927	0.9937

BiModal :

Classification Models	Accuracy	Precision	Recall	F1-Score
LSTM Model	0.6752	0.6169	0.5564	0.5851
Bidirectional LSTM Model	0.6743	0.6073	0.5902	0.5986
LSTM CNN Model	0.7783	0.7819	0.6846	0.7300
Bidirectional LSTM CNN Model	0.7632	0.7725	0.6685	0.7167

6.3 Observations

In this section we state the important observations from the above results.

For Single Modal:

LSTM-CNN Model with glove embeddings gives best accuracy as well as f1-score while K-Nearest Neighbour Classifier gives least accuracy and f1-score.

For BiModal:

LSTM-CNN Model gives best accuracy and f1-score while standalone LSTM and Bidirectional LSTM Models without CNNs give poor accuracy.

Therefore, We find LSTM-CNN as the best fit for the model and select it for identifying the news as real or fake in both single modal and bimodal approach to fake news detection.

7. CONCLUSION AND FUTURE SCOPE

To conclude, in this paper, we have proposed a single modal as well as bimodal fake News Detection System using either text (single modal) or text and images both(bimodal). The results obtained in single modal are extremely good and for bimodal as well the results are quite good. We have achieved these results with the help of combining the powers of CNN and LSTM Models and creating a hybrid CNN-LSTM Model.

While these results are satisfactory, there is always room for improvement. Thus in future, instead of having a classifier trained on a single category of data like political news in our case, we can build a more general model trained on various categories of information like sports news, technology news, etc. which can broaden the news detection capabilities.

Additionally, instead of having a static model which requires data to be provided beforehand, we can have a dynamic model which is continuously updated with newer data and thus continues to learn new things.

8. REFERENCES

[1] Monther Aldwairi, Ali Alwahed. "Detecting Fake News in Social Media Networks",in the 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2018).

[2] Muhammad Umer, Zainab Imtiaz, Saleem Ullah ,Arif Mehmood, Gyu Sang Choi,Byung-won On "Fake News

Stance Detection Using Deep Learning Architecture (Cnn-lstm)",Ieee Access,2020.

[3] Julio C. S. Reis, A. Correia, Fabricio Murai, Adriano Veloso, F. Benevenuto, E. Cambria "Supervised Learning for Fake News Detection",Published 2019,IEEE Intelligent Systems.

[4] Majed Alrubaian, Muhammad Al-Qurishi, Mohammad Mehedi Hassan "A Credibility Analysis System for Assessing Information on Twitter",IEEE Transactions on Dependable and Secure Computing.

[5] Shivam B. Parikh; Pradeep K. Atrey "Media-Rich Fake News Detection: A Survey", 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR).

[6] Kuai Xu; Feng Wang; Haiyan Wang; Bo Yang "Detecting fake news over online social media via domain reputations and content understanding",Published 2020,Tsinghua Science and Technology.

[7] Shivam B. Parikh; Vikram Patil; Ravi Makawana; Pradeep K. Atrey "Towards Impact Scoring of Fake News",2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR).

[8] Saad Sadiq; Nicolas Wagner; Mei-Ling Shyu; Daniel Feaster "Supervised Learning for Fake News Detection",2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR).

[9] Sahil Chopra, Saachi Jain, John Merriman Sholar. "Towards Automatic Identification of Fake News: Headline-Article Stance Detection with LSTM Attention Models".

[10] Sarthak Jindal,Raghav Sood,Richa Singh,Mayank Vatsa,Tanmoy Chakraborty. "NewsBag: A Multimodal Benchmark Dataset for Fake News Detection".

[11] Kai Nakamura, Sharon Levy, William Yang Wang. "r/Fakeddit:A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection".

[12] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun,Kishlay Jha, Lu Su,Jing Gao "EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection".

[13] Tanushree Mitra, Eric Gilbert. "CREDBANK: A Large-Scale Social Media Corpuswith Associated Credibility Annotations".

[14] Jiawei Zhang, Bowen Dong, Philip S. Yu. "FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network".

[15] Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, Vasudeva Varma. "MVAE: Multimodal Variational Autoencoder for Fake News Detection"

[16] Thanh Thi Nguyen, Quoc Viet Hung Nguyen, Cuong M. Nguyen, Dung Nguyen, Duc Thanh Nguyen,Saeid Nahavandi. "Deep Learning for Deepfakes Creation and Detection: A Survey"

[17] Fei Yan, Krystian Mikolajczyk "Deep correlation for matching images and text"

[18] Pritika Bahad, Preeti Saxena, Raj Kamal "Fake News Detection using Bi-directional LSTM-Recurrent Neural Network".

[19]Ahmed H, Traore I, Saad S. "Detecting opinion spams and fake news using text classification", Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018. 2.

[20]Ahmed H, Traore I, Saad S. (2017) "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham (pp. 127-138).