iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VI **Month of publication:** June 2022

DOI: https://doi.org/10.22214/ijraset.2022.44623

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

News Aggregator with Fake News Detection using Stacked LSTMs

Nishant Racherla[1], Bhavesh Kankani[2], Adarsh Reddy Pulakanti[3], Talakoti Mamatha[4], Vasundhara Bejjam[5]
[1, 2, 3, 4, 5]*Dept. of Computer Science and Engg. Sreenidhi Institute of Science and Tech, Yamnampet, Hyderabad, 501301*

Abstract: ***In a post-truth world, fake news has spread globally in equal proportion. From industrialized countries like the United States, Norway, and Ireland to emerging ones like India, Brazil, and others, no one appears to be immune. Because India is the world's largest democracy with the second largest population, it is particularly vulnerable to fake news. Low literacy rates, combined with an avalanche of fake news, make it difficult to carry out the true spirit of democratic decision-making, putting the country's democracy at risk. Using a Long Short-Term Memory (LSTM) network, our system presents a way of detecting and eliminating fake news from different sources. The news items are also tagged and delivered to the user according to their choices.***
Keywords: News, Fake News, Machine Learning, LSTM, Neural Network, Aggregation

I. INTRODUCTION AND OVERVIEW

With the drastic increase in Internet access, the amount of people hopping on to social media has also seen a surge. Because of the easy availability and exponential expansion of information available on social media networks, distinguishing between false and real information has become difficult. The ease with which information may be shared has aided in the exponential expansion of information deception. Where the propagation of false information is common, the credibility of social media networks is also at risk.
The circumstances mentioned above raise the need for a reliable system, that is, the system that is able to provide the user with an unbiased result stating whether the given news article is fake or real. Manually combing through each article and determining if it is phony or authentic is a time-consuming and laborious endeavor. Machine Learning proved to be a gamechanger in circumstances like these, analyzing almost thousands of training and testing data from datasets to produce a virtually flawless model to detect false news. To categorize news items into fake and real categories, this research employs machine learning and data science principles.
The system's aggregator feature connects users to relevant sources of information, allowing them to learn more about the news. This is accomplished by aggregating numerous related news stories based on tags.

II. LITERATURE REVIEW

In the recent past, many papers were published dealing with the topic of fake news detection. One such paper by Mykhailo Granik and Volodymyr Mesyura [1] explored the idea of using a Naïve Bayes classifier for classifying news into fake and real classes. A Naïve Bayes classifier is a probabilistic model which implements the Bayes theorem from probability. The accuracy they achieved by testing their model was 74%. Though Naïve Bayes classifiers are typically used for solving classification problems, for the current problem this level of accuracy is not satisfactory. A linguistic analysis approach was suggested by DSKR Vivek Singh and Rupanjal Dasgupta in their paper [2], but their approach reached an accuracy of 87%.
Another paper by Hadeer Ahmed, Issa Traore, and Sherif Saad [3] suggested an approach that involved a Linear Support Vector Machine classifier. This model performed feature extraction using TF-IDF and achieved an accuracy of 92%. Although this approach has pretty good accuracy, it is not suitable for datasets that are large and contain noise.
An approach that involved the use of a Long Short-Term Memory model [4]. LSTM models work well with sequential data. The training dataset consists of time-series data, which is an extension of sequential data. This approach achieved an accuracy of 91%.
Taking into consideration all the aforementioned approaches and their drawbacks, the model proposed in this paper, which is a Stacked LSTM model, strives to achieve much higher accuracy in comparison with other models.

III. SCRAPING

Web Scraping is a technique used to extract text from web pages. Automated web scraping can help extract news content from news articles of various news websites. Use of web scraping can prove to be helpful by extracting news articles from trusted news companies' websites and updating the database with real news. This can help keep track of the current happenings and eventually help in training the model to detect for any fake news spread on social media sites.

IV. DATASETS

Two datasets are used to train the Machine learning model. One dataset is ISOT fake news dataset compiled by professors from the University of Victoria. The dataset consists of 21,417 real news articles and 23,481 fake news articles gathered from various sources. Each article data consists of the title, text, type and date on which the article was published. The other dataset is obtained by making use of the Twitter API. User must have a developer account registered in Twitter Developer Platform to gain access to the API keys and tokens. Using the credentials available, the data is scraped with the help of a web scraper. The dataset obtained by scraping consists of 14,012 rows of data. Each row is a data entry with 4 columns namely id, text, source and class.

V. BACKGROUND KNOWLEDGE

A. Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a form of Deep Learning Neural Network which can effectively analyze the organized representations of data in the form of arrays. A CNN's potential is brought out in detecting design components like lines, gradients, circles, and even images (For example, eyes and faces) when given as inputs. This property makes Convolutional Neural Networks extremely powerful in the field of Computer Vision. An underexposed picture can directly be fed to a CNN to obtain desired results without any pre-processing. A Convolution Neural Network has a similar architecture to a Feed-Forward Network containing 20 layers. The core of the Convolution Neural Network that differentiates it from a Feed-Forward Neural Network is the Convolution Layer. This powerful layer makes a CNN more accurate and efficient compared to a typical Feed-Forward Neural Network.

A Convolutional Layer is capable of detecting increasingly complex shapes or data. Multiple Layers are stacked upon each other to form a CNN. Depending on the complexity of the problem at hand and the data available for training, the CNN architecture can be modified to suit the problem thereby increasing the chances to obtain a model which gives desired results. For example, to detect handwritten numbers only three or four convolution layers are sufficient. But human face detection may require more than 25 layers.

The main objective of a CNN is to develop a machine learning model which is able to perceive and react to the environment as humans do. This is done by incorporating common sense into the model. This makes CNN ideal to perform operations such as image and video recognition, image inspection and classification, media recreation, recommendation systems, natural language processing, and so on.

In short, a convolutional neural network is a multi-layered feed-forward neural network obtained by stacking numerous hidden layers on top of each other by following a particular order. The sequential nature of a CNN makes it easy to learn and effectively recognize the hierarchical properties of data. The architecture of a typical CNN consists of grouping and hidden layers followed by convolutional layers which are in turn followed by activation layers. This architecture is inspired by the analogous network of neurons in the human brain [5].

B. Recurrent Neural Networks

RNN is a strong and robust form of neural network. It is one of the emerging algorithms with high potential and is extensively in use due to its unique property being its internal memory [6]. During its time of release, RNNs were said to be the only neural networks with internal memory. Recurrent neural networks are relatively new compared to many other machine learning methodologies. They were developed in the 1980s but were unable to get proper recognition. RNN gained popularity in the 1990s as a result of increased computer power, vast quantities of data to deal with, and the advent of long short-term memory (LSTM) in the 1990s.

The internal memory of RNN is used to recollect important data about the input received, hence, boosting the accuracy of predicting the occurrence of the next event. This property makes it an ideal choice for time series, speech, text, financial data, audio, video, weather, and a variety of other sequential data types. Recurrent Neural Networks in comparison with other Deep Learning algorithms have greater scope to learn about a sequence and its environment.

Sequential data represents data that is arranged in such a way that related elements are placed successively. Examples include financial data and DNA sequences. Time series data being one of the most commonly used forms of sequential data is defined as a collection of data points in chronological sequence.

The information in an RNN is processed in loops. The layers consider current input and knowledge obtained from the previous input which proves to be beneficial in obtaining desired results. An RNN after processing an input, duplicates the output, where one copy moves forward in the network while the other is stored in the internal memory of the layer for future reference. Thus, at any given instance of time, we observe two inputs for the RNN layer, present input, and data from the recent past. This is a vital feature of RNN because the data contains crucial information about the next input or data in the sequence, therefore RNN is able to accomplish such tasks effectively that other algorithms cannot.

C. Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are an extension of Recurrent Neural Networks (RNN) developed to address the limitations of RNN. RNNs networks operate on current inputs while taking past outputs (feedback) into consideration and storing them in memory for a limited period of time (short-term memory) [5]. Hence, RNNs are unable to store data for an extended period of time.

RNNs have a number of drawbacks that need to be dealt with. To begin with, while dealing with time-series data, there are cases where the algorithm may require data that was generated a long time ago in order to predict the current result accurately. Such "long-term dependencies" are difficult to handle using RNN alone. Secondly, there is no better control over which background elements should be brought forward and how much of the past should be "forgotten". Other shortcomings of RNN include vanishing and explosive gradients which are uncovered during the backtracking phase of the network.

Owing to these drawbacks, LSTM networks came into existence. The approach of LSTM resulted in nearly complete elimination of vanishing gradient without affecting the logic of the training model. LSTMs can handle noise, dispersed representations, and continuous data and are used to bridge lengthy time gaps in some problems.

In contrast to Hidden Markov Model (HMM), preserving a finite number of states from the start is not required. LSTMs provide a variety of parameters, including learning rates, input and output biases, and learning rates. As a result, no precise changes are necessary. The complexity of updating each weight is reduced to O(1) with LSTMs, which is equivalent to Back Propagation Through Time (BPTT).

The major difference between the RNN and LSTM is the hidden layer of LSTM being a unit or gated cell. The hidden layer comprises of four layers that interact to give the cell's output and state. These are passed to the next hidden layer. LSTM has three logistic sigmoid gates and one tanh layer whereas an RNN has just one layer of tanh. The sigmoid acts as a filter to limit the amount of data that could pass through a cell. They play a major role in deciding whether the incoming data is required by the next cell and are free to reject the data otherwise.

VI. PROPOSED SYSTEM

A. Stacked LSTM for Fake News Detection

The system proposed in this paper implements a Stacked LSTM model. A typical LSTM model comprises of single hidden LSTM layer followed by a standard feed forward output layer. The Stacked LSTM is an extension of the LSTM approach where the model comprises of multiple hidden layers and each layer is a collection of memory cells. Stacking LSTM hidden layers makes the model deeper, more accurately earning the description as a deep learning technique. Thus, Stacked LSTM models have a greater chance of success on a wide range of challenging problems due to the depth of the network.

The additional layers play a major role in recombining the learned features and knowledge from previous layers and generate new predictions in the later stages of the model. This increases the scope of data interpretation and analysis. The increase in the depth of the network reduces the number of neurons required per layer and reduces the time to train the model to perfection. Graves et al. presented stacked LSTMs or Deep LSTMs in their use of LSTMs to voice recognition, beating a benchmark on a difficult standard issue. To model talent, they discovered that the depth of the network was more essential than the number of memory cells in a given layer in the same study.

B. Aggregator Module

The news articles that are to be displayed to the user are of 8 different categories – All News, India, Business, Science, Technology, Entertainment, Sports, and Health in the form of tabs. The user can select news category preferences to their liking when they sign up. The “All News” tab displays the news of the user’s preferred categories. All of these news articles are obtained from a public API called “NewsAPI”, which is a reputed source for a multitude of news websites. These articles then undergo tag-based aggregation and are rendered in their respective tabs.

VII. IMPLEMENTION

For the implementation of the proposed system, Python, an open-source programming language is used. The implementation consists of three stages – Pre-processing, ML model generation, and ML model training. Pre-processing and ML model generation is done using various modules/frameworks of Python.

A. Pre-processing

Pre-processing is the process of modifying the raw data, which comprises noise, missing values, and may also be in an incorrect format. The ability of a Machine Learning model to learn depends mainly on the dataset with which it is trained. Hence, pre-processing is considered to be a fundamental step and is to be done before feeding the data into the ML model. The fake news detection problem involves providing a lot of text into the model. A Machine Learning model cannot process raw text and learn from it. Hence, there is a need to convert raw text into something the model can digest and learn. This pre-processing is done in two stages – dataset pre-processing and text pre-processing.

1) Dataset pre-processing: The dataset used to train and test the model is a culmination of two separate datasets: one consisting of only real news data and the other consisting of only fake news data. These two datasets undergo cleaning and are merged to form the final dataset which is fed to the model. In the final dataset, the real news is represented by a binary value of 1, and the fake news is represented by a binary value of 0.
2) Text pre-processing: The Text pre-processing required for the model consists of Tokenization, Removal of unnecessary punctuation and tags, Removal of stop words, Stemming/Lemmatization of text, and feature extraction. The use of Natural Language Processing techniques can be helpful for pre-processing the text.
3) *Tokenization:* Tokenization deals with the separation of a large piece of text into smaller units called tokens. The true intention of tokenization is to create a vocabulary on which a model can rely for understanding the corpus of text as a whole. To enhance the performance of the model, the text is tokenized to create a vocabulary out of top K frequently occurring words [7].
4) *Removing Unnecessary Punctuation:* Punctuation tags, special characters, and emojis present in the human-written text are a source of noise while training the model [8]. For example, the tokens ‘congrats’ and ‘congrats!’ have the same meaning but are treated as two separate entities which are redundant. For this purpose, the removal of unnecessary punctuation tags, and special characters becomes necessary.
5) *Removing Stop Words:* Text contains many words which have a meaning when read by humans but are considered to be useless when processed by a Machine Learning model. Such words collectively are known as stop words [7]. For example, the article ‘the’ is a stop word as it does not actually change the meaning of the sentence to a substantial degree. Removal of stop words before feeding the text into the model can obviate excess processing time of the model.
6) *Stemming/Lemmatization:* Text consists of a lot of inflected words. Stemming and Lemmatization techniques aid in trimming these inflected words into root words for achieving text normalization before feeding the text into the ML model [9]. For example, ‘act’ and ‘acting’ are words originating from the same root. Hence, trimming the word ‘acting’ to ‘act’ can help obviate the excess processing time of the model. The main difference between stemming and lemmatization techniques is that stemming of an inflected word may not result in an actual word, whereas in the case of lemmatization of an inflected word actual word is obtained. For example, stemming of ‘relieving’ results in ‘reliev’ (the actual root is ‘relieve’), and lemmatization of ‘relieving’ results in ‘relieve’.
7) *Feature Extraction:* The words after undergoing tokenization, removal of punctuation & stop words, and stemming/lemmatization are still in the text form. The next step is to encode this text into real-valued vectors which can be directly fed into the ML model. This mapping of textual data to real-valued vectors is known as Feature Extraction. Word Embedding models can be used for serving this purpose. One such Word Embedding model is the “word2vec” model [10].

B. Model Generation

A Sequential model is generated using Keras, a high-level API of TensorFlow. The first layer in the model is an Embedding layer which helps in the conversion of each index representing a word into vectors of fixed size. Next comes LSTM layers which help in learning the model. The last layer is a dense layer with a sigmoid activation function for removing the less fired neurons from determining the output.

C. Model Training

A Deep Learning model is trained by adjusting the weights of each node in the network in order to reduce the error made by a model over the training dataset. A typical deep learning model consists of large neural networks and it takes a large amount of time to train the model when compared to other Machine Learning models [11]. Hence, CuDNN layers are used which help in reducing the training time of the model.

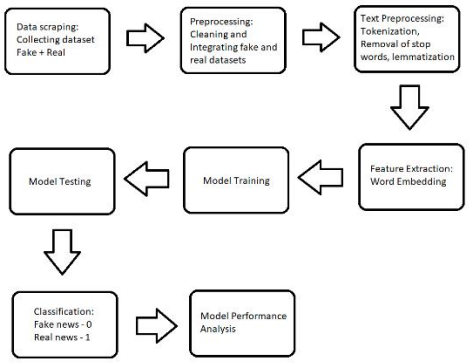


Fig. 1: This flowchart shows the workflow of the proposed ML model.

VIII. RESULTS

A. *Performance Metrics*

1) *Accuracy:* The accuracy of a model can be defined as the ratio of correct predictions to the total predictions performed by the model. The accuracy achieved by the model implemented, after training and testing is 98.52%.

```
[ ]  1  accuracy_score(y_test,y_pred)

     0.9852115812917595
```

2) *Confusion Matrix:* A confusion matrix can be defined as a table mapping the predictions of the test data fed into the machine learning model to their respective categories being TP (True-positives), TN (True-negatives), FP (False-positives) and FN (False-negatives). Various other metrics can be derived from the confusion matrix. The name "confusion" is given to the matrix because the matrix effectively alleviates any confusions regarding the performance and reliability of the model [12]. Table 1 displays the confusion matrix of the proposed model.

TABLE I. CONFUSION MATRIX

Confusion Matrix		Actual Values	
		YES	NO
Predicted Values	YES	5816 (TP)	21 (FP)
	NO	69 (FN)	5319 (TN)

3) *Precision:* The ratio of accurately categorized positive samples (True Positive) to the total number of classified positive samples is referred to as precision [13]. The precision score of the proposed model is 0.9960.

```
1  from sklearn.metrics import precision_score
2  precision_score(y_test, y_pred)

0.9960674157303371
```

4) *Matthew's Correlation Coefficient:* This coefficient is used to evaluate classifier models. It measures the difference between observed and predicted classifications [14]. Its value lies between -1 and +1. If the coefficient is +1, the model can be considered perfect. The coefficient for the model proposed in this paper is 0.9839.

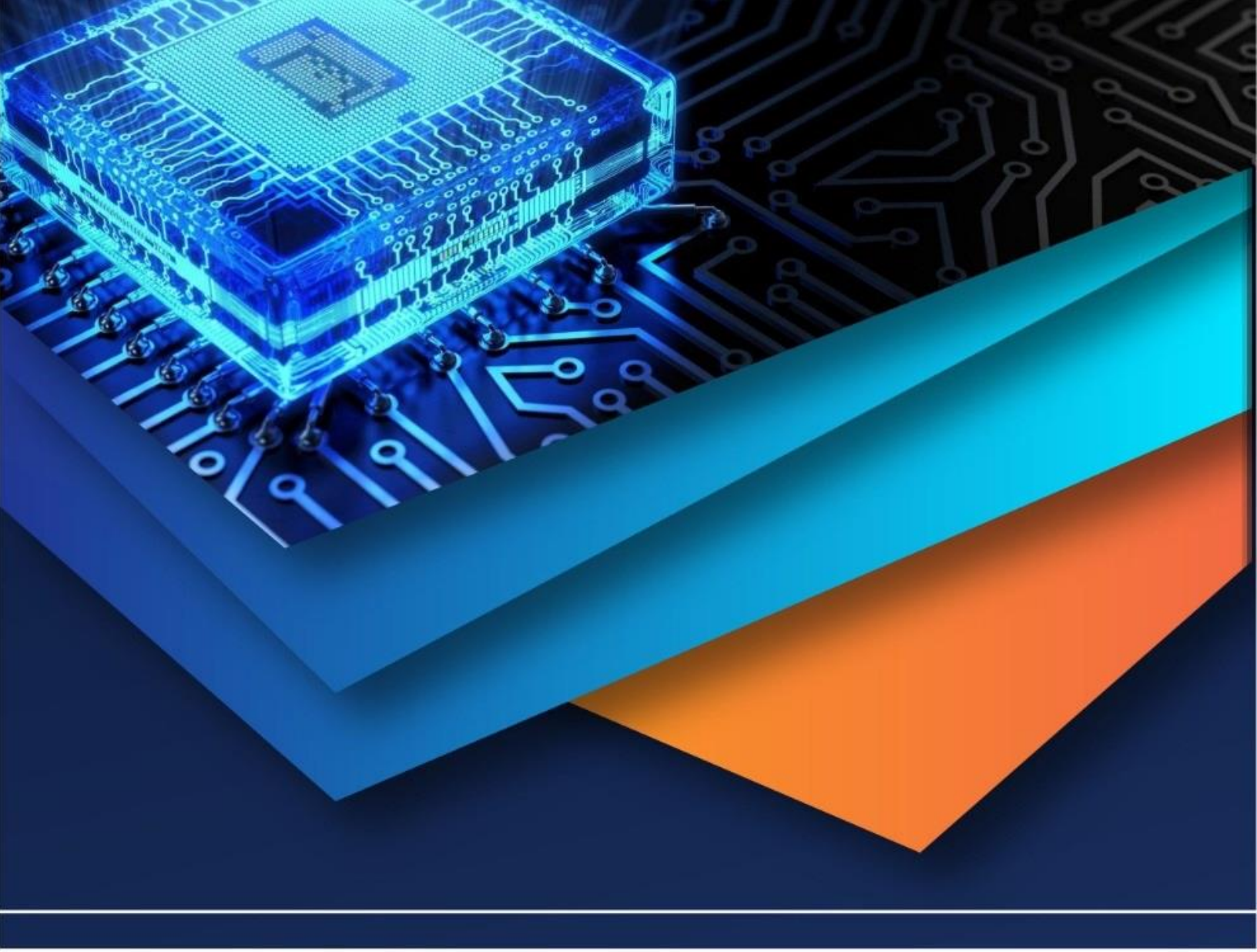
```
1  from sklearn.metrics import precision_score
2  precision_score(y_test, y_pred)

0.9960674157303371
```

IX. ACKNOWLEDGMENT

We would like to thank our college, Sreenidhi Institute of Science and Technology for providing us the valuable opportunity to write research papers under highly qualified and experienced guides.

REFERENCES

[1] Mykhailo Granik and Volodymyr Mesyura. Fake news detection using naive bayes classifier. In 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), pages 900–903. IEEE, 2017.
[2] DSKR Vivek Singh and Rupanjal Dasgupta. Automated fake news detection using linguistic analysis and machine learning.
[3] Hadeer Ahmed, Issa Traore, and Sherif 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, pages 127–138. Springer, 2017.
[4] Tejaswini Yesugade, Shrikant Kokate, Sarjana Patil, Ritik Varma, Sejal Pawar. Fake News Detection using LSTM. International Research Journal of Engineering and Technology (IRJET), Volume 8 Issue 4, 2017.
[5] Pritika Bahad, Preeti Saxena, Raj Kamal, Fake News Detection using Bi-directional LSTM-Recurrent Neural Network, Procedia Computer Science, Volume 165, 2019, Pages 74-82, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.01.072.
[6] Sherstinsky, Alex. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena. 404. 132306. 10.1016/j.physd.2019.132306.
[7] S. Pradha, M. N. Halgamuge and N. Tran Quoc Vinh, "Effective Text Data Preprocessing Technique for Sentiment Analysis in Social Media Data," 2019 11th International Conference on Knowledge and Systems Engineering (KSE), 2019, pp. 1-8, doi: 10.1109/KSE.2019.8919368.
[8] Kaur, Dupinder. (2017). Sentimental Analysis on Apple Tweets with Machine Learning Technique.
[9] P. Han, S. Shen, D. Wang and Y. Liu, "The influence of word normalization in English document clustering," 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), 2012, pp. 116-120, doi: 10.1109/CSAE.2012.6272740.
[10] Ma, Long & Zhang, Yanqing. (2015). Using Word2Vec to process big text data. 10.1109/BigData.2015.7364114.
[11] M. Sato, “Performance comparison of LSTM with and without cuDNN(v5) in Chainer.” https://chainer.org/general/2017/03/15/Performance-of-LSTM-Using-CuDNN-v5.html. Accessed: 2018-05-04.
[12] Hossin, Mohammad & M.N, Sulaiman. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. International Journal of Data Mining & Knowledge Management Process. 5. 01-11. 10.5121/ijdkp.2015.5201.
[13] J. A. Cottam, N. C. Heller, C. L. Ebsch, R. Deshmukh, P. Mackey and G. Chin, "Evaluation of Alignment: Precision, Recall, Weighting and Limitations," 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 2513-2519, doi: 10.1109/BigData50022.2020.9378064.
[14] D. Chicco, V. Starovoitov and G. Jurman, "The Benefits of the Matthews Correlation Coefficient (MCC) Over the Diagnostic Odds Ratio (DOR) in Binary Classification Assessment," in IEEE Access, vol. 9, pp. 47112-47124, 2021, doi: 10.1109/ACCESS.2021.3068614.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY