Predicting Rumors in Social Media Using RNN and NLP

Chayana Rajput[1], Ankit Garg[2]
[1] *PG Student, Neelam College of Engineering and Technology*
[2]*Assistent Professor, Neelam College of Engineering and Technology*

Abstract-The spread of rumors is quickly becoming one of the most significant issues that sectors and the business world face today. It is possible for pervasive dissemination of false information to have devastating consequences, not only for individuals but also for society as a whole. The objective of disseminating false information via rumors is to deceive audience into believing that it is true. Due to this, identifying rumors based on the content of individual articles can be difficult and time-consuming. In this paper, we propose a technique for predicting rumors in social media using RNN and NLP to circumvent this issue. Initially, datasets are collected from publicly accessible sources. After the data has been collected, the pre-processing stages are executed, which include text normalization, lemmatization, and the removal of stop words. Tokenization is the reduction of a sentence or paragraph into its constituent terms. Word embedding is achieved through the use of POS, a procedure that entails assigning a tag to each word. Recurrent Neural Networks (RNN) used to classify the data. Finally, it has been determined whether the news was genuine or a rumor the entire time. The results indicate, the proposed strategy performs well when compared to the performance of alternative deep learning strategies.

Index Terms—-Deep Learning, Fake News Classification, Information Verification, Lemmatization, Natural Language Processing (NLP), Recurrent Neural Network (RNN).

I. INTRODUCTION

It is not a recent phenomenon, nor is it something that can be easily resolved; in fact, it has been practised all over the world for many years under a variety of titles, such as propaganda or rumours. Rumors, known as junk news, is the deliberate dissemination of false information, hoaxes, or rumours through old-style print and media with broadcast news, as well as online social media and articles published on news media websites [1]. People, organisations, and even countries may be impacted by rumors which have been compared to a virus due to its contagious nature and tendency to spread by itself. From a historical perspective, rumors presents ever since Gutenberg established the printing press, which led to the development of printed media and the growth of the news. This was one of the most unforgettable occasions. As we can see, the phenomenon known as rumors goes by a number of various labels; yet, they all refer to the same thing, which is the deception of the individuals who read it. It's possible to divide false information into two categories: knowledge-based and intention-based. Misinformation and disinformation are two subcategories that can be further subdivided under the intentional category. Misinformation is the unintended sharing of erroneous information that is based on the beliefs and point of view of an individual who disseminates it [2].

Since the dawn of human civilization, there has always been something passing itself off as legitimate news. However, the proliferation of false information is facilitated by recent technological developments as well as the transformation of the global media environment. It's possible that the spread of false news will have enormous repercussions for the social, political, and economic contexts [3]. There are many different manifestations of bogus information and rumors. Rumors have a significant influence on our perspective of the world since information shapes that perspective. Because of the information, we are able to make important judgments. When we learn more about a topic or a group of individuals, we are better able to form an opinion about them. If we find information that is created, phoney, twisted, or otherwise not accurate on the internet, we will not be able to make sound judgments [4].

The identification of false information spread via social media presents a number of novel and difficult research challenges. The proliferation of user-generated content on the internet and in social media led to the proliferation of rumors, which poses a threat to the established standards of journalism. The following is a synopsis of the issues that have been brought about by rumors [5]. In the first place, the purpose of rumors is to trick consumers into reading material that is intentionally dishonest and difficult to spot, all for the sake of making money. The vast majority of the time, this content consists of a variety of subject matters and writing styles, as well as linguistic methods for making it an actual news. To find a solution to this issue, the additional Information is utilized at our disposal, namely, meta-data, social interaction, and so forth. The data quality is the second issue that arises as a result of utilising this new information. Because rumors relies on news' short bursts and timed events, it follows that these events will not be checked in an accurate manner by knowledge bases or specialists. As a result, a large amount of information that is unstructured and noisy is being created, leading to a problem with big data [6].

The typical approach to resolving this problem is to request assistance from experts, such as journalists, who are tasked with examining assertions in light of evidence derived from facts that have been said or published earlier. Nevertheless, it is a costly and time-consuming endeavour. For instance, in order to determine whether or not a piece of news is genuine, PolitiFact1 uses the opinions of three editors [9]. Automatic detection of false news is being developed with the intention of cutting down on the amount of time and effort required by humans to identify rumors and assisting us in putting an end to the spread of such stories. With the advancement in many fields within Computer Science, the problem of identifying false news has been investigated from a variety of angles [10].

The proliferation of e-readers has been facilitated by the ease with which they may connect to the internet and by the devices' user-friendly interfaces. These factors contributed to a steady rise in the dissemination of false information throughout social media platforms and other websites. A speedily growing way of identifying fraudulent information using machine learning algorithms may be done with the use of NLP. The consumption of news can be seen as a double-edged sword. People are more likely to seek out and consume news due, on the one hand, to the fact that it is inexpensive, easy to acquire, and rapidly disseminates information [11].

The topic of detecting rumors is a brand-new research subject; thus, before beginning research on the problem, it is essential and important to provide a formal explanation of the problem as well as to formulate it. The textual information that pertains to the news stories, their producers, and the subjects of those pieces may be culled from the various online social media platforms. This information includes their profiles and descriptions. It will be necessary to have an efficient feature extraction and learning model in order to collect signals that show their level of believability. Furthermore, as previously discussed, the credibility labels of news items, creators, and subjects are inextricably linked. These correlations may be represented by the authorship and article-subject linkages that exist between the various entities. A successful inclusion of such connections into the learning framework would be useful for producing outcomes that are more precise about the legitimacy of false news [13].

To overcome these issues, a technique for predicting rumors in social media using RNN and NLP has been proposed. The main objectives are to efficiently detect the rumors using natural language processing and deep learning techniques, to differentiate between original and rumors using headline or the content of the article without using labelled source content provider, and to eliminate the need for intensive data engineering or transformation work.

II. LITERATURE REVIEW

Tschiatschek et al. [15] utilized the Facebook added in Germany for reporting rumors to fight it. By using the users' feedback on the trustworthiness of news (through the use of flags), they made a procedure called Detective. Bayesian inference is used by this algorithm to find rumors, and it learns over time how accurate users' flags are.

Kim et al. [16] also created a similar strategy, although they placed more of a focus on the trade-off that exists between flagging news and exposing people to false information. They design an algorithm called CURB that chooses which articles should be sent for fact checking before they are exposed to a sufficient number of people. This allows them to control not just which stories are fact verified but also when those stories are fact reviewed. Their algorithm demonstrated the ability to successfully curb the circulation of fabricated news and inaccurate information.

Liu et al. [17] focus on the limits that occur from recognising false news by using news characteristics and news dissemination factors. This is done in order to get around the problems that arise from attempting to detect rumors. In particular, they take into account the manner in which information spreads throughout the network, which is modelled as a multivariate time series. Within this model, each tuple is a numerical vector representing the characteristics of a user who participated in the dissemination of the information.

Yang et al. [18] take advantage of this and extract hidden latent characteristics both in pictures and in text using their method. They were able to tackle the challenge of detecting false news by combining characteristics from these two sources, which resulted in the creation of TI-CNN. This network was trained in a unified vector space.

The approaches that were based on posts studied the comments, thoughts, and responses of people as potential signs of false news. These features included things like comments, views, feelings, ratings from users and tags. The network-based approaches utilised by social networking sites allowed for the quick dissemination of false information. These methodologies attempted to develop and study networks from a variety of different viewpoints. For example, friendship networks investigated the interaction between users and their followers. On the other hand, stance networks are representations of commonalities between posts. Another kind of network is the co-occurrence network, which analyses the degree to which a person and a subject are relevant [21].

III. SYSTEM ANALYSIS

3. 1 PROPOSED METHODOLOGY

3.1.1 OVERVIEW

Initially, data preprocessing is performed in which the unstructured raw data gathered from a range of sources are transformed in order to get the data ready for analysis. The building of an NLP model necessitates both the preprocessing of text and the factorization of data. The regular expression, lemmatization, tokenization, and removal of stop words are just some of the processes that are applied to each row of the news title and text in order to recover the root keywords. Another process that is utilized is tokenization, which breaks down each word into its component parts. The 1DCNN algorithm is already well-known, and it is utilized frequently in text mining as well as image analysis. For most text classification and natural language processing (NLP) jobs, a Conv1D, also known as a one-dimensional convolution neural network, is the network of choice.

3.1.2 DATASET COLLECTION

The identification of instances of false news serves as the dataset for this research project (Kaggle.com). There is a wide variety of online resources, such as Kaggle and UCI Machine Learning that provide information on current events. This research intends to develop a deep learning model that is capable of differentiating between real and fake news simply by looking at the headline or the content of the article, even in the absence of a labelled source content provider as a part of its ultimate objective. English-language content that is freely accessible online and originates from the United States is a good choice for datasets, particularly on websites such as Kaggle. This type of content is particularly useful for competitions. In addition to this, they feature clear labels, which remove the necessity for extensive amounts of data engineering or transformation labour.

3.1.3 DATA PREPROCESSING

In any process involving Machine Learning, the step known as "Data pre-processing" is the one in which the data undergoes a transformation to bring it to a state in which it can be easily passed by the machine. As the data set is compiled from various sources, the first step is to purge it of any superfluous or irrelevant information by changing it to lowercase, removing punctuation and symbols, and eliminating stop words. The content from social media that has been extracted for the purpose of fact checking needs also be subjected to some degree of transformation and pre-processing before a model can be constructed using that content. The building of an NLP model necessitates both the pre-processing of text and the tokenization of data.

1. Text Normalization

The procedure of changing text in a canonical form is referred to normalisation of the text. When text is normalised before it is stored or processed, it allows the separation of the data needed from remaining data, which enables the system to transmit constant data as an input to the subsequent phases of the procedure. Normalizing the text also makes it possible to process or store the text.

2. Stop Word Removal

Words like articles, prepositions, conjunctions, and so forth are examples of stop words. The removal of stop words is a type of pre-processing phase that involves removing these stop words and, as a result, assisting in the subsequent steps and decreasing some processing time due to the significant reduction in the size of the document that occurs as a result.

3. Lemmatization

During Text Mining, lemmatization, also known as stemming, is a stage in the pre-processing phase. Stemming is also a relatively typical need of functions related to Natural Language processing. In point of fact, it is considered to be of utmost significance in most of the information retrieval systems. The reduction of a word's many grammatical forms and word forms to its root form is the primary objective of the stemming process. In order to achieve its purpose, stemming seeks to reduce all inflectional forms and sometimes other forms of a word that are connected to its derivation to a single base form. A stemmer for the English language, for example, should recognise the strings "cats," "catlike," and "catty" as being derived from the root "cat."

4. Tokenization

To tokenize means to break a sentence, paragraph, and/or any other type of text into individual words. In addition to doing extremely fundamental pre-processing operations on the text, such as removing all of the punctuation marks, it tokenizes the content. The number of times each word appeared in the document is tallied as an integer, and this information is delivered in the form of an encoded vector along with the length of the full vocabulary.

For example: Kerry to go to Paris in gesture of sympathy

How Kerry lead differently in Paris

Table 1 Output for Tokenization

Output	Kerry	go	to	Paris	in	gesture	sympathy	Of	lead	How	Differently
0	1	1	2	1	1	1	1	1	0	0	0
1	1	0	0	1	1	0	0	0	1	1	1

5. Word Embedding

Word vector embedding is a representation of words that is based on the notion that words that are utilized in the same context tend to have meanings that are comparable to one another. It depicts each word as a high-dimensional dense vector, and, with regard to the hypothesis, it employs comparable vectors for words that appear in contexts that are comparable to one another. There are a number of different ways to get such a representation, some of which include dimensionality reduction, neural networks, and probabilistic models, amongst others. It is possible to train it using the processing data, or it can be loaded from a previously trained embedding matrix. One potential drawback of embedding models is that each word, even one that can have multiple interpretations, will have just one vector assigned to it. An expansion that provides additional vectors for a single word that can have numerous meanings, one for each of the word's distinct meanings, is used to solve the problem. In order to use an embedding, the documents that are being inputted first need to be converted into a sequence of integers. First things first: compiling a comprehensive list of each and every word that appears in the corpus. The second step is to construct a dictionary that links every term to its respective index in the dictionary.

The BERT training model is a deep learning system that has demonstrated cutting-edge outcomes across many NLP applications. This model is utilised as the training model. The pre-training language knows that BERT was developed by Google. BERT is a powerful word-embedding design that was developed on a transformer-encoded framework. It has already been pre-trained. The ability of the BERT method to recognise and record the contextual importance of a text makes it stand out from other approaches.

3.1.4 RNN

Previous studies have shown that many popular machine learning algorithms, including Support Vector Machines (SVMs), Random Forests (RFs), and Naive Bayes (NBs), have subpar detection accuracies. We also assess the performance of other deep learning algorithms, including CNN. Although these systems have improved, a new modified deep learning algorithm based on RNN is required for improved fake news detection.

RNNs are a type of neural network in which the same unit of computation (sometimes termed a "cell") is used many times to interpret a variety of inputs. Figure 2 (a) depicts this pattern of repetition visually.

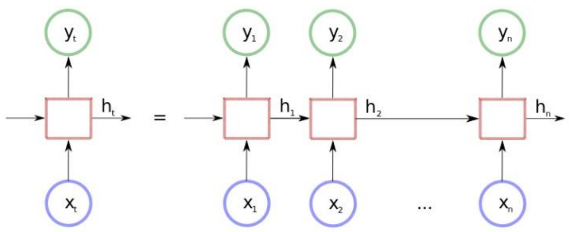

Figure 3.1.4 (a) Recurrent Neural Network

To calculate the new hidden state and the output at time step t, we need the input,y_t and the previous step's hidden stateK_{t-1}.

$$k_t = \sigma_k(W_k y_t + V_k K_{t-1} + c_k) \quad (1)$$

$$x_t = \sigma_x(W_x K_t + c_x) \quad (2)$$

Where:
At instant t, the input vector is denoted by y_t
The hidden layer vector is denoted by K_t
At time step t, x_t is the output vector.
W, Uc are parameter matrices and vectors, respectively.
σ_k, σ_x represent activation functions

As a direct consequence of the prior arts, the Long Short-Term Memory Network, which is essentially an RNN that has been altered to include gating mechanisms, has been developed in order to circumvent the limitations that are inherent to conventional RNNs?

The disadvantage of conventional RNNs is addressed by inserting three gates into each network cell to support the idea of memory. Each time the cell reads an input, a memory is created and modified.

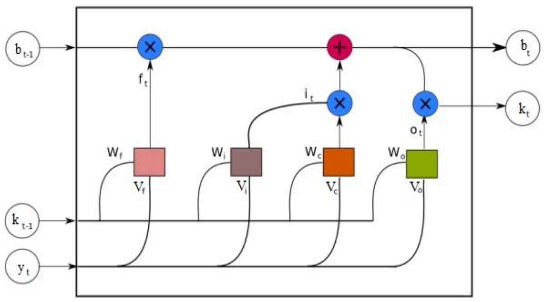
Figure 3.1.4 (b) illustrates LSTMs with four gate

$$b_t = f_t * b_{t-1} + i_t * \widetilde{b_t} \quad (3)$$

Forget gate: determines pieces of information should be removed from currently stored ones in memory. When an input is provided at y_t time step t, the answer is computed as follows:

$$f_t = \sigma(W_f y_t + V_f K_{t-1} + c_f) \quad (4)$$

b_{t-1} then gets multiplied with this f_t for transforming with some information that is removed.

Memory gate: creates a new candidate for the memory. It is computed as follows, given that y_t is an input:

$$\widetilde{b_t} = \text{tank}(W_b y_t + V_c K_{t-1} + b_c) \quad (5)$$

Input gate: This gate determines the quantity of information of the candidate memory will be injected into the updated one. Given an input y_t , it is computed as:

$$i_t = \sigma(W_i y_t + V_i K_{t-1} + b_i) \quad (6)$$

$\widetilde{b_t}$is then multiplied by it in order to obtain the newly added memory that will be stored in new memory cell.

Output gate: defines the depth of extraction of the memory from the cell. Calculated as:

$$O_t = \sigma(W_o y_t + V_o K_{t-1} + b_o) \quad (7)$$

After that, the newly hidden state is updated as follows:

$$K_t = o_t * \sigma_b(b_t) \quad (8)$$

The solution to the issue of long-term dependencies is supposedly found in the existence of internal memory, which has the capacity to receive updates in a sequential fashion.

The method known as the Attention Mechanism enables selective focus to be placed on various aspects of the input that is being received. The attention mechanism was initially implemented in machine translation, where it yielded remarkable results.

3.1.5 OVERALL ALGORITHM

The overall algorithm of the proposed work is given below.

Step 1: Dataset is collected from the open source.

Step 2: Once the data is collected, the pre-processing such as text normalization, lemmatization, and stop word removal is performed.

Step 3: Tokenization is done to break down a paragraph or sentence into words.

Step 4: Word embedding is done using POS, which is process of assigning a tag to each word.

Step 5: The classification is performed through RNN.

Step 6: Finally, the news is identified whether it is real or fake.

IV. RESULTS

4.1.RESULTS AND DISCUSSION

The proposed Fake news detection using deep learning and NLP is implemented on Jupyter Notebooks.

4.1.1 DATASET USED

The dataset used in the proposed experiment is given below.

https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset?select=True.csv

The detection of fake news was the dataset that is used for this study (Kaggle.com). There are numerous websites, including Kaggle and UCI Machine Learning that contain data on the news. This data set is all about Real or Fake News or Text dataset. It contains number, title, text, and label.

4.2 RESULTS

Importing the dataset and showing it in tabular form as given in Table 2.

Table 2 Dataset

	title	text	class
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	0
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	0
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	0
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	0
...	...	...	...
44893	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	1
44894	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	1
44895	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	1
44896	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	1
44897	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	1

We drop the unnamed column since we do not need it for further predictions.

1. Label Values as we can see the data in the dataset

```
(23481, 21417)
```

Figure 4.2(a) Label Values

2. Applying Stop words to remove the reoccurring words in the English language.

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\91858\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Figure 4.2(b) Applying stopwords

3. Snowball stemmer is used for stemming.

4. Regular expression library is used to remove the punctuation and other unnecessary symbols.

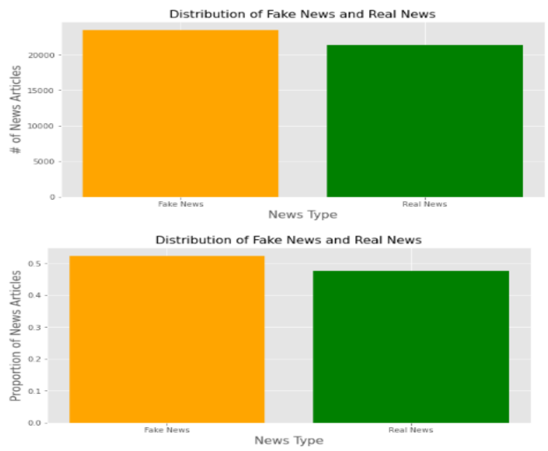

Figure 4.2(c) EDA

```
▾ MultiLabelBinarizer
MultiLabelBinarizer()
```

Figure 4.2(d) Applying Multilabel Binarizer

Count vectorizer and Tfidf vectorizer are used for performing tokenization.

```
▾ CountVectorizer
CountVectorizer(binary=True, max_features=2000, ngram_range=(1, 2))
```

Figure 4.2(e) Count Vectorizer

```
▾ TfidfVectorizer
TfidfVectorizer(max_features=2000, ngram_range=(1, 2))
```

Figure 4.2(f) Tfidf Vectorizer

4.1.3 PARAMETER CALCULATION OF ML MODELS

The performance metrics such as accuracy, recall, precision, F1-score, time to train, time to predict, and total time taken are estimated for each ML models. In addition, confusion matrix for each model is defined.

1. Naïve Bayes (Multinomial NB)

	precision	recall	f1-score	support
0	0.93	0.96	0.94	7712
1	0.95	0.92	0.93	7105
accuracy			0.94	14817
macro avg	0.94	0.94	0.94	14817
weighted avg	0.94	0.94	0.94	14817

Figure 4.1.3(a) Performance Metrics for Naïve Bayes (Multinomial NB)

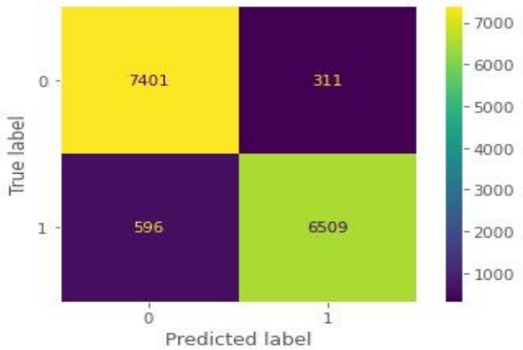

Figure 4.1.3(b) Confusion Matrix for Naïve Bayes (Multinomial NB)

2. Linear SVC

	precision	recall	f1-score	support
0	0.96	0.96	0.96	7712
1	0.96	0.96	0.96	7105
accuracy			0.96	14817
macro avg	0.96	0.96	0.96	14817
weighted avg	0.96	0.96	0.96	14817

Figure 4.1.3(c) Performance Metrics for Linear SVC

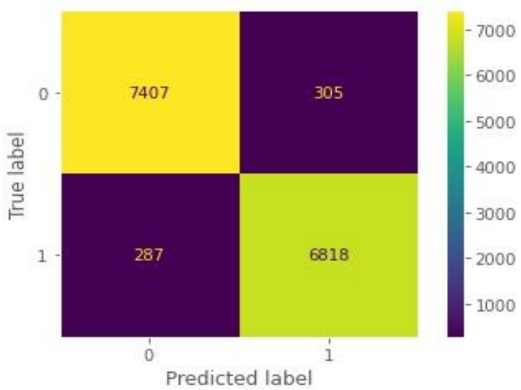

Figure 4.1.3(d) confusion matrix for linear svc

4.1.3 IMPLEMENTATION OF RNN

The RNN is implemented as follows. Initially, the layers are defined.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, None, 128)         1280000

 bidirectional (Bidirectiona  (None, None, 128)        98816
 l)

 bidirectional_1 (Bidirectio  (None, 32)               18560
 nal)

 dense (Dense)               (None, 64)                2112

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 1,399,553
Trainable params: 1,399,553
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.1.4(a) Implementation of RNN

```
Epoch 1/10
1078/1078 [==============================] - 468s 428ms/step - loss: 0.2440 - accuracy: 0.8839 - val_loss: 0.0542 - val_accurac
y: 0.9841
Epoch 2/10
1078/1078 [==============================] - 448s 415ms/step - loss: 0.0493 - accuracy: 0.9864 - val_loss: 0.0297 - val_accurac
y: 0.9891
Epoch 3/10
1078/1078 [==============================] - 445s 413ms/step - loss: 0.0263 - accuracy: 0.9936 - val_loss: 0.0330 - val_accurac
y: 0.9903
Epoch 4/10
1078/1078 [==============================] - 451s 419ms/step - loss: 0.0128 - accuracy: 0.9971 - val_loss: 0.0270 - val_accurac
y: 0.9908
Epoch 5/10
1078/1078 [==============================] - 451s 419ms/step - loss: 0.0099 - accuracy: 0.9979 - val_loss: 0.0339 - val_accurac
y: 0.9919
Epoch 6/10
1078/1078 [==============================] - 445s 412ms/step - loss: 0.0054 - accuracy: 0.9990 - val_loss: 0.0516 - val_accurac
y: 0.9805
```

Figure 4.1.4(b) Running the Model

Prediction of Accuracy will be calculated as 0.9906458797. The performance metrics such as precision, recall, F1-score, and support are estimated.

```
Accuracy on testing set: 0.9906458797327394
Precision on testing set: 0.9863362667901806
Recall on testing set: 0.9941643323996265
```

Figure 4.1.4(c) Accuracy prediction

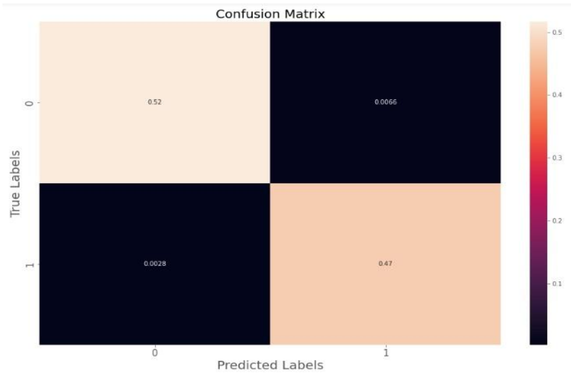

Figure 4.1.4(d) Confusion Matrix

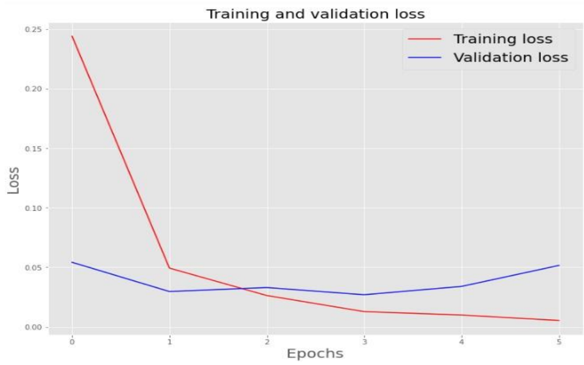

Figure 4.1.4(e) Training and Validation Loss graph

It is concluded that the proposed RNN has more accuracy when compared to the other ML models. Hence, the rumors are detected accurately.

V. CONCLUSION

The proposed work include the detection of rumours (fake news) using deep learning and NLP. Initial datasets are collected from public sources. Text normalization, lemmatization, and stop word removal are performed after data collection. Tokenization means, dividing a paragraph or sentence into individual words. POS is used for word embedding, which is the process of designating a tag to each word. The classification is accomplished using RNN. Finally, the authenticity of news is determined. The results demonstrate that the proposed work outperforms other deep learning techniques. Incorporating additional features comprising meta-data of news content is a further application of the feature engineering method. This is one way the method can be utilized to its fullest capacity.

REFERENCE

[1]. Conroy, N. J., Rubin, V. L., and Chen, Y. (2015). Automatic deception detection: Methods for finding rumors. Proceedings of the Association for Information Science and Technology, 52(1):1–4.

[2]. Deligiannis, N., Do, T. H., Nguyen, D. M., and Luo, X. (2018). Deep learning for geolocating social media users and detecting rumors.

[3]. Khurana, U. and Intelligentie, B. O. K. (2017). The linguistic features of rumors headlines and statements. Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[4]. Kirilin, A. and Strube, M. (2018). Exploiting a speaker's credibility to detect rumors. In Proceedings of Data Science, Journalism & Media workshop at KDD (DSJM'18).

[5]. Long, Y., Lu, Q., Xiang, R., Li, M., and Huang, C.-R. (2017). Rumors detection through multi-perspective speaker profiles. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers), volume 2, pages 252–256.

[6]. S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent Features of Rumor Propagation in Online Social Media," in Proceedings - IEEE International Conference on Data Mining, ICDM, pp. 1103–1108, 2013.

[7]. Mihalcea, R. and Strapparava, C. (2009). The lie detector: Explorations in the automatic recognition of deceptive language. In Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 309–312. Association for Computational Linguistics.

[8]. Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[9]. Mitra, T. and Gilbert, E. (2015). Credbank: A large-scale social media corpus with associated credibility annotations. In ICWSM, pages 258–267.

[10]. Nakashole, N. and Mitchell, T. M. (2014). Language aware truth assessment of fact candidates. In Proceedings of the 52nd Annual Meeting of the Association for 6093 Computational Linguistics (Volume 1: Long Papers), volume 1, pages 1009–1019.

[11]. Eugenio Tacchini, Gabriele Ballarin, Marco L. Della Vedova, Stefano Moret, and Luca de Alfaro. Some like it hoax: Automated rumors detection in social networks.

[12]. James Thorne, Mingjie Chen, Giorgos Myrianthous, Jiashu Pu, Xiaoxuan Wang, and Andreas Vlachos. Rumors stance detection using stacked ensemble of classifiers. In Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism, pages 80–83, 2017.

[13]. Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S. Yu. Ti-cnn: Convolutional neural networks for rumors detection.

[14]. K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake News Detection on Social Media: {A} Data Mining Perspective," CoRR, vol. abs/1708.0, 2017.

[15]. S. Tschiatschek, A. Singla, M. Gomez Rodriguez, A. Merchant, and A. Krause, "Fake News Detection in Social Networks via Crowd Signals," in Companion Proceedings of the The Web Conference 2018, WWW '18, (Republic and Canton of Geneva, Switzerland), pp. 517–524, International World Wide Web Conferences Steering Committee, 2018.

[16]. J. Kim, B. Tabibian, A. Oh, B. Schoelkopf, and M. Gomez-Rodriguez, "Leveraging the Crowd to Detect and Reduce the Spread of Fake News and Misinformation," 2017.

[17]. Y. Liu and Y.-F. Brook Wu, "Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks," 2018.
[18]. Y. Yang, L. Zheng, J. Zhang, Q. Cui, Z. Li, and P. S. Yu, "{TI-CNN:} Convolutional Neural Networks for Fake News Detection," CoRR, vol. abs/1806.0, 2018.
[19]. K. Popat, S. Mukherjee, A. Yates, and G. Weikum, "DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning," CoRR, vol. abs/1809.0, 2018.