Comparative Analysis of Fake News Detection using Machine Learning and Deep Learning Techniques.

[1]Nerissa Pereira, [2]Simran Dabreo, [3]Linnet Rodrigues, [4]Prof. Merly Thomas

[1, 2, 3]Students, [4]Professor

[1]Department of Computer Engineering, University of Mumbai,

[1] Fr. Conceicao Rodrigues College of Engineering, Mumbai, India.

Abstract: In recent times there has been a tremendous increase in the spread of false information. The issues related to the spread of fake news across multiple social media platforms as well as random web pages from the internet has gained massive importance from the modern day journalism due to its emerging popularity in various research communities. The intention behind designing and spreading the fake news content is to misguide the readers and make them believe the false news. In our daily lives, it is a difficult task to distinguish the fake content from the news articles; as most of the fake articles tend to be perfectly structured to make the readers believe their content. Hence fake news cannot be classified solely based on the content, but we also need to consider multiple attributes such as the source of the news, the user engagements, the authenticity of the user sharing the news, etc. In this paper we have come up with the applications of NLP and Neural Networks techniques for detecting the 'fake news'. Generating a model based only on a count vectorizer (using word tallies) or a (Term Frequency Inverse Document Frequency) tfidf matrix, (word tallies relative to how often they're used in other articles in your dataset) we can only get a certain accuracy; as they will not consider the interdependency between a variety of features present in the news content; as well as the order of the content. For this, we propose our system using a deep learning unit called as LSTM combined with neural networks and try generating the comparative analysis.

Index Terms - Social Media, Twitter, NLP, Neural Networks, Deep Learning Models, RNN, LSTM.

I. INTRODUCTION

The motive behind creating a fake news is basically to mislead people by making them fall prey to a variety of hoaxes, propaganda and inaccurate information. There are articles that are either completely false or just random opinion pieces presented as news. Some of the major platforms for the spread of fake news include Facebook, Twitter, Whatsapp, Reddit, etc. Often, false news try to mimic real headlines and manipulate the content. In today's world most of the organizations employ the use of social media and hence own the accounts on Twitter, Facebook and Instagram for multiple business purposes such as announcing corporate information, advertisement regarding new product releases, etc. Consumers, investors, and other stake holders take these news messages as seriously as they would for any other mass media. Detecting misleading information on social media is an extremely important but also a technically challenging problem. The difficulty arises from the fact that even the human eye cannot accurately distinguish true from false news. The prevalence of Fake news enforced a great impact over the entire democracy during the US Presidential Elections held in the year 2016. Media later stated that since a lot of US adults tend to use social media to get news, it was definitely the reason why Donald Trump was favoured over Hillary Clinton.In India the spread of fake news has reached a new peak since 2019 with multiple events, from the general elections and Pulwama attack to scrapping of Article 370 and the ongoing protests against the Citizenship Amendment Act, as well as the recent crisis which has been taking a hit due to the outspread of Corona Covid-19 virus. All these events triggering lead to a massive spread of misleading news and opinions across multiple social media platforms.

In this paper we propose a method for identification of the fake news using a few Machine Learning algorithms such as Naïve-Bayes, Support Vector Machine (SVM), Logistic Regression and Deep learning algorithms like LSTM's and Neural Networks using Keras.

Our project consists of two major sections: In the first section we perform comparative analysis between different algorithms viz. Naïve-Bayes, SVM, Neural Networks using Keras, and Recurrent Neural Networks (LSTM). In the second part we retrieve the real time tweets from twitter and classify them as suspicious and non-Suspicious based on text as well as user characteristics using the Logistic Regression Classifier.

II. LITERATURE REVIEW

Many of existing fake news detection techniques are based on the process of feature extraction. Linguistic Based methods use key linguistic features from the fake news. Some of these features include N-Grams, Punctuations, Readability and Syntax. In this paper, Shivam B. Parikh, Pradeep K. Atrey, (2018) [1] have implemented a Naïve-Bayes classifier based on the concept that fake news articles often use the same set of words while true news will have a particular set of words. The overall accuracy score of the model using naïve-bayes classifier was around 70%. The accuracy of the simple naïve-bayes classifier can be improved by combining it with the n-gram model. This paper concludes by presenting some of the open research challenges such as verifying the source of the news article, determining the credibility of the author of the news and so on.

The paper by Ajao, Bhowmik and Zargari (2018) [3] proposes a method using hybrid RNN models. Recurrent Neural Networks have shown considerable accuracy in time and sequence based predictions. LSTM recurrent neural network (RNN) was adopted for the sequence classification of the data. Two other variations of LSTM namely *LSTM with dropout regularisation* and *LSTM*

with CNN were also implemented. The analysis has been performed on the LIAR dataset. LSTMs preserve the memory from the last phase and incorporate this in the prediction task of the neural network model. The accuracy using the plain vanilla LSTM was found to be around 82%; while the LSTM-CNN hybrid model performed better than the dropout regularization model, with a 74% accuracy In terms of precision and recall, it has been observed that the plain vanilla LSTM model achieves the best performance. The size of the dataset also plays an important role in improving the accuracy score of the model. Most of the Deep learning models such as RNN often use multiple neural network layers in order to train the model effectively. Hence, it has been concluded that the dataset should be of a considerable size.

Another model for analysing the tweets from twitter was constructed by Sheryl Mathias, Namrata Jagadeesh (2019) [7]. In this paper they have stated that the most prominent content based features were number of unique characters, swear words, pronouns and emoticons in a tweet; and user based features included the number of followers and length of username. The classifier models were trained on the dataset which contained 250 tweets. The results were calculated using various models with Random Forest achieving the accuracy of 74%, Logistic Regression achieving the accuracy of 76% and Decision Tree giving the lowest accuracy of 67%. The major drawback was that the model was not able to analyse the categories of words since it did not involve the usage of n-gram model. Also the dataset did not consider any information of the user who posted the tweets such as the username, follower counts, friends count, etc. If the user features are also considered in addition to the tweet information, the model will tend to achieve better accuracy scores.

III. COMPARATIVE ANALYSIS

This will be our first level of implementation in which we will perform the classification on the dataset using four different algorithms. The result will be displayed for each algorithm with its individual accuracy score. We will be using the following four algorithms: Naïve-Bayes, Support Vector Machine, LSTM using Recurrent Neural Networks and Keras Neural Network.

3.1 Workflow Diagram:

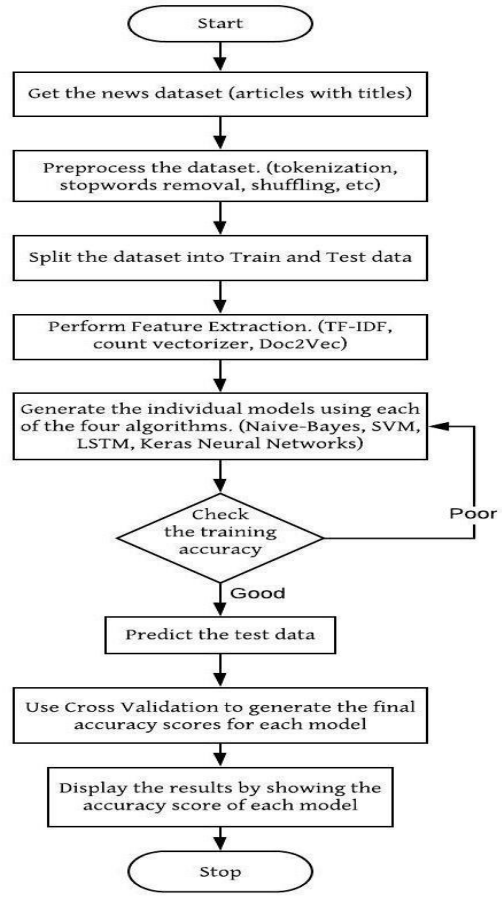

Figure 1: Flowchart for Comparative Analysis

3.2 Dataset:

The dataset used to test the efficiency of the model is produced by kaggle, containing 25000 news articles. News categories included in this dataset were business; politics, science and technology, entertainment, and health. This dataset obtained from Kaggle was noisy and required cleaning. The main features included in each row of the data were id, title, author, text, classification of being "fake" or "real . We spilt the corpus into training and testing dataset. The training data has 20000 corpus of news articles and the testing data has 5000 corpus of news articles.

3.3 Data Pre-Processing

In this section we present various pre-processing techniques applied on the data from the dataset. The dataset downloaded from kaggle had a lot of raw data including missing rows and columns. Thus we first need to process this data and then split it into train and test data set. Some of the common data processing techniques which we have used in our project include: stopwords removal, stemming, tokenization and POS tagging.

Stemming: In Natural Language Processing, Stemming can be referred to as a process of reducing different forms of a word to a core root. Words that are derived from one another can be mapped to a central word or symbol, especially if they have the same core meaning. This works as an information retrieval setting and boosts the algorithm's recall.
Tokenization: Tokenization is a process in which the sequence of string is broken into pieces such as keywords, words, phrases, symbols, etc. called as tokens. In the process of tokenization, some characters like punctuation marks are discarded. All characters within contiguous strings are part of the token.
Part-Of-Speech Tagging (POS): The POS tagging is a process of assigning corresponding part of speech like noun, verb, adjective, verb to each word present in the sentence. The tag may indicate one of the parts-of-speech, semantic information and so on. In our project POS tagging is applied to language grammatical rules to parse meanings of sentences and phrases present in our news corpus.

4. Feature Extraction
In this section we have enlisted the various feature selection methods which we have performed using sci-kit learn python libraries.
Count Vector: The Count Vectorizer provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary. An encoded vector is returned with a length of the entire vocabulary and an integer count for the number of times each word appeared in the document.

TF-IDF: The Term Frequency is used to summarize how often a given word appears within a document. Inverse Document Frequency is used to downscale words that appear a lot across documents. It also allows you to encode new documents.
a) A vocabulary of words is extracted from the news article and each word is assigned a unique integer value in the output vector.
b) We then calculate the inverse document frequencies for each word in the news article, and assign a lowest value of 1.0 to the word which occurs most frequently.
c) Next, we encode the article using n-element sparse array and we can view the scores of each word having different values from the other words present in the document.
We then normalize these scores by mapping them to a value between 0 and 1. Thus the document vectors are now encoded and can be used as an input to our four different classifiers.

3.5 Classifiers used to train our model:

A) Naïve-Bayes: In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes theorem with strong (naive) independent assumptions between the features. Bayes classifiers evaluate every feature independently, which means it does not consider the interdependency between the values of two or more features. It typically use bag of words features to identify spam news, a commonly used approach in text classification. The main idea is to treat each word of the news article independently. It considers the fact that, fake news articles often use the same set of words, which may indicate, that the specific article is indeed a fake news article.
We consider the following formula for calculating the conditional probability of the news:
$$Pr(F|W)=Pr(W|F)\cdot Pr(F)/(Pr(W|F)\cdot Pr(F)+Pr(W|T)\cdot Pr(T)), \quad (1)$$

Pr(F|W) – conditional probability, that a news article is fake given that word W appears in it;

Pr(W|F) – conditional probability of finding word W in fake news articles;

Pr(F) – overall probability that given news article is fake news article;

Pr(W|T) – conditional probability of finding word W in true news articles;

Pr(T) – overall probability that given news article is true news article.

B) Support Vector Machine: A support vector machine (SVM), is also considered to be a supervised learning algorithm. SVMs work by being trained with specific data already organized into two different categories. Hence, we construct a model after it has already been trained. Furthermore, the goal of the SVM method is to identify which category any new data falls under, as well as, it must also maximize the margin between the two classes.

C) LSTM: Long short term memory is an extension of the RNN (Recurrent Neural Network). In addition to RNN, LSTM's also have memory over the long run. It comprises of three gates namely input gate, output gate and forget gate. The forget gate is used to forget features that have little value or weight. As the algorithm keeps running, it learns what is important and what is not by assigning the weights accordingly. This characteristic makes it the best fit for our news data, as the corpus is extensively large and we need to eliminate the unwanted data to predict the right label on the text. We feed our news article content to the LSTM unit in the vector embedded form, after it has been tokenized into words. For our project we have used the Sequential LSTM model. Each word is given to a separate LSTM unit. The basic LSTM unit is as follows:

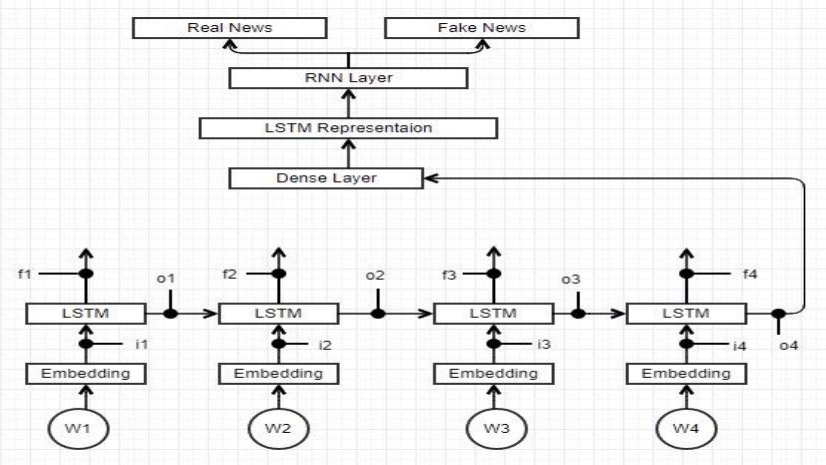

Figure 2: Working of LSTM

In Figure 2, (i1, i2, i3) represent the input gates; (o1, o2, o3) represent the output gates; (f1, f2, f3) represent the forget gates and (w1, w2, w3) represent the tokenized words from the news articles.

1. Every word of the news article is embedded into a vector form and given as an input to the LSTM unit through the input gate.
2. The LSTM unit calculates the weight of the word and produces an output via the output gate. This output is then concatenated with the input of the next LSTM unit. This unit now calculates the combined weight of the two words. If a particular word or phrase is found to be redundant, the LSTM unit discards it via the forget gate. The important features evaluated are stored in the memory of LSTM in order to identify the similar features from the next input more efficiently.
3. This process is carried out sequentially, till the entire news article is evaluated. The output is then passed to the dense layer which summarizes the received output and converts it into a proper LSTM representation.
4. This representation is then passed on to the RNN layer. The RNN layer also receives other features such as the title, author and label from the dataset. Based on these features along with the LSTM encoded news article, it evaluates the functions and classifies the news as Real or Fake.

D) Neural Networks using Keras: Artificial Neural Networks are also commonly called Feed forward neural networks. Feed forward neural networks are relatively simpler than the recurrent neural networks. These networks are named so because the information moves linearly in the network through the input layer, then the hidden layer and ultimately through the output layer. In our project we implement one feed-forward neural network model using Keras. Keras is a popular open-source neural-network library written in Python. It contains several procedures of frequently used neural-network building blocks such as layers, objectives, and activation functions. In our project neural network implementation uses three hidden layers. In the Keras implementation we use layers of size 256, 256, and 90 along with dropout layers. The ReLU, which is also known as Rectified Linear Unit is used as the "activation function".

IV. CLASSIFICATION OF REAL TIME TWEETS FROM TWITTER

This will be our second level of implementation in which we retrieve the tweets from twitter and classify them as suspicious or non-suspicious based on various text as well as user characteristics. Twitter is a popular micro-blogging service, which has gained resurgence as one of the prominent news source and information dissemination agent over the recent years. Every post on Twitter is defined mainly by two main components: the tweet (content and associated metadata) and the user (source) who posted the tweet. The content on twitter is characterized by a maximum tweet length of 140 characters.

Some of the most common predictive variables for determining fake news are: number of original tweets, number of retweets, average length of original tweets, retweets, etc. These features combined with linguistics can help uncover words and phrases which indicate whether a tweet will be perceived as highly credible or less credible.

As seen in our above mentioned first level of implementation that Neural Networks using LSTM's have achieved the highest accuracy score; but while classifying the real time tweets from twitter, we use the logistic regression algorithm to train the model. This is because the tweets on twitter have a maximum character length of 140 and hence not much context can be extracted from those tweets. As LSTM is an extension to the implementation of Recurrent Neural Networks, it needs a huge subject based corpus for training, owing to which it is not suitable for classifying the real time tweets. So for twitter news prediction we will proceed with logistic regression.

The tweets present in our dataset are related to various events such as the Las Vegas shooting, Hurricane Harvey that occurred in Houston, and so on. We have a total of 500 tweets in our dataset, out of which 188 are labelled as fake and 312 are labelled as real tweets.

4.1 Classifier:

Logistic Regression: For the classification of our real-time tweets extracted from twitter, we have used the Logistic Regression classifier. Various text as well as user features extracted are given as an input to the classifier. Logistic Regression is a supervised machine learning algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (true, success, etc.) or 0 (false, fail, etc.). It uses gradient descent to converge onto the optimal set of weights (Θ) for the training set. We have used the following sigmoid function for our project:

$$h_\theta(x) = g(\theta^T x) = 1/\ 1+exp(-\theta^T x)$$

4.2 Implementation

The flow of the model is shown in figure 3.

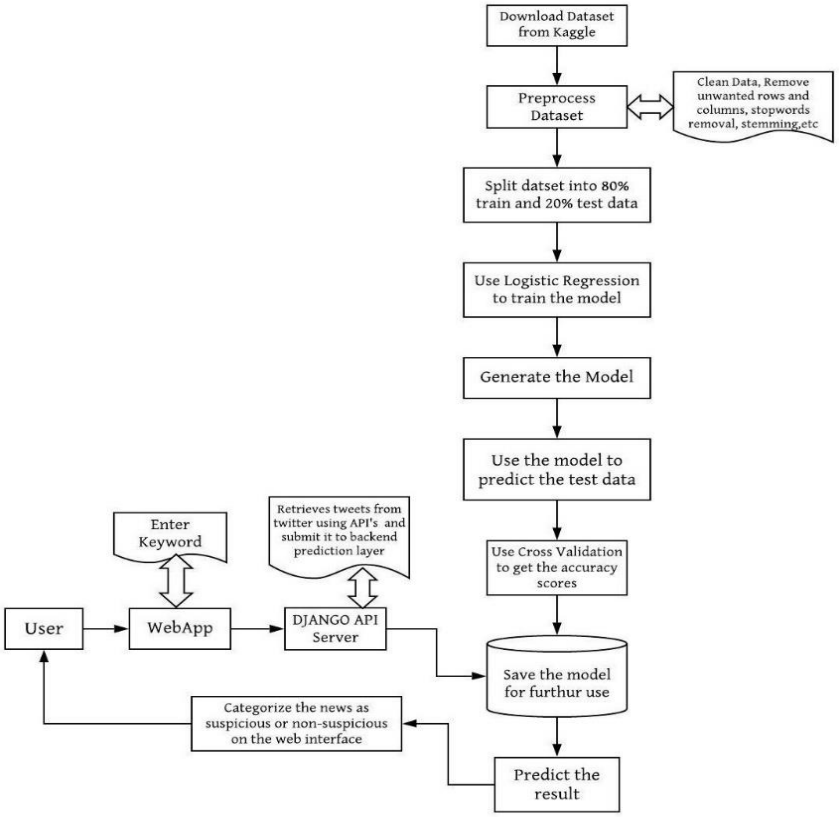

Figure 3: Flowchart for classification of Real Time Tweets from Twitter

The step wise implementation is as follows:

1. To extract the tweets from twitter, create an App on twitter's developer account and then download the consumer keys and access tokens. These keys are then embedded into the code to retrieve the tweets.
2. We Then perform web scraping to retrieve a certain amount of tweets. Many features are extracted for determining real and fake news. These features can be identified as user and tweet features.

- Following are the tweet features in our dataset:
  - Text
  - Created at
  - Retweet count
  - Favourite count
  - Source
  - Length

- Following are the user features in our dataset:
  - User id
  - User name
  - User created at
  - User description
  - User follower count
  - User friends count
  - User status count
  - User verified
  - User status count
  - User profile contains a URL.

3. We then divide the dataset into training and testing data by splitting the dataset in the ratio of 80:20 for applying logistic regression algorithm.
4. The text of the tweets are tokenized using NLTK package. We then eliminate the stopwords present in the tweets. We used regex package in Python to count the number of hashtags, mentions, question marks and exclamation marks respectively.
5. Later, we cleaned the text of the tweets by removing URLs and punctuation marks using regex package and then counted the number of characters and words in the tweet.
6. All of the above extracted features are then passed on to the logistic regression classifier. The probabilistic co-efficient value for every feature is calculated and finally the label is assigned as "REAL" or "FAKE". The coefficient value generated for each feature plays an important role in predicting the label on the tweet.

7. We now feed the test data to the classifier and predict the labels on the test data. The predicted values are then compared to the actual labels of the dataset to measure accuracy of the classifier.

V. RESULTS

In this section we discuss the results by calculating the accuracies of the various models mentioned in the comparative analysis. The accuracy, precisions and f1 score can be computed with the help of confusion matrices.
We have constructed a graph shown in Figure 4 by taking different algorithms on X-axis and accuracy on the Y-axis. It is inferred that LSTM provides us with the highest accuracy of 93%, followed by Keras Neural Network achieving an accuracy of 90.3%, SVM achieving an accuracy of 85% and finally Naïve Bayes achieving the lowest accuracy of 68%. The Keras based neural network has a good accuracy that almost matches LSTM. LSTM provides such high results because it uses the concept of memory which saves the important features/text from the current news and uses it to establish a relationship with the next input. Also it uses the back propagation to improve the accuracy after every epoch. The values shown are the averaged values over successive trials.

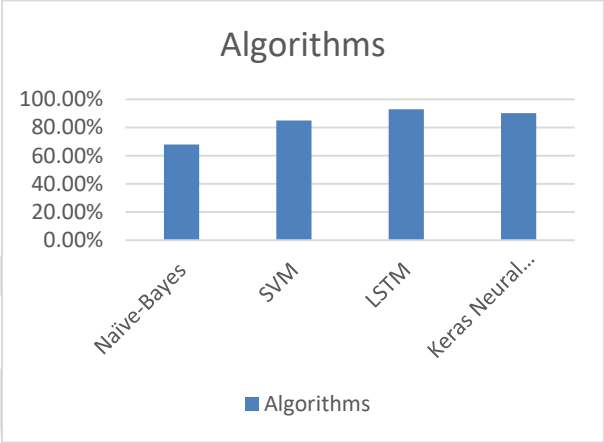

Figure 4: Comparison of Algorithms

In the level 2 of implementation we have classified the real time tweets from twitter as suspicious and non-suspicious using the supervised learning logistic regression algorithm. The accuracy of the model over the training dataset having 16 parameters was 97.2% whereas the accuracy of the model on the testing dataset with same parameters was found to be around 87%. Using LSTM to classify the real time tweets dropped down the accuracy to around 79%; due to the restriction on the length of the tweets. Following graphs show the accuracy of the model built using logistic regression to classify the real time tweets.

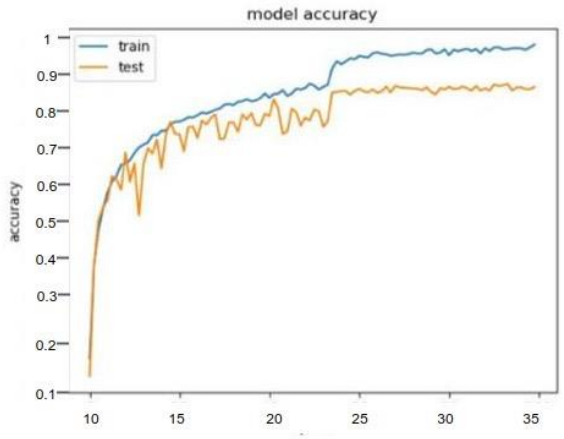

Figure 5: Training and Testing Accuracy for logistic regression on real time tweets

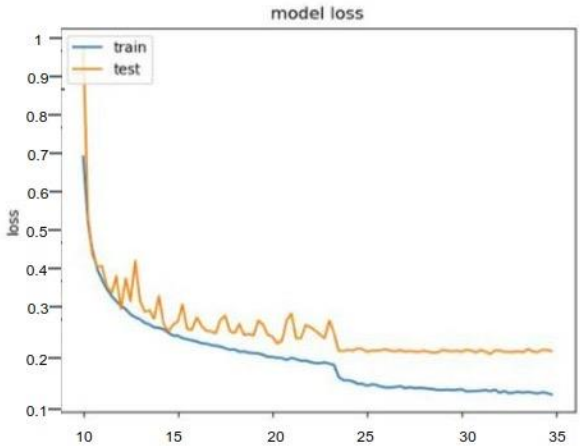

Figure 6: Model Loss for logistic regression on real time tweets

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a model for fake news detection using a variety of machine learning and deep learning algorithms. Furthermore, in the first level of implementation, we investigated the four different classifiers and compared their accuracies. The model that achieves the highest accuracy is LSTM and the highest accuracy score is 93%. Fake news detection is a quite popular and trending research area which has an extremely scarce number of datasets. The current model which we have generated is run against the existing dataset, indicating that the model performs well against it.
In our next level we have analysed the real time data from twitter. Here we have trained our model using logistic regression algorithm; due to the inability of the LSTM to perform well over real time tweets having considerably small length. The accuracy for the tweets classification using Logistic Regression was found to be around 87%. A future scope for the real time tweets classification would be to use n-gram models and combine it with the classifiers to achieve a better accuracy and precision. Another limitation is the size of the twitter dataset which contains only 500 tweets. This is a very small sample size, which drastically affects the accuracy scores for the test data. Future scope would be to increase the sample size to at least 1000 tweets.
Visual presentation also plays a huge role in people believing in fake news. Hence in the future work we need to verify not just the language but also the images and audio embedded in the content.

VII. ACKNOWLEDGEMENT

We would hereby like to express our gratitude to our Mentor, Prof. Merly Thomas for her exceptional support without whom this paper and the research behind it would have not been possible. We have been truly benefited by her valuable thoughts, suggestions and ideas. We thank her for her constant words of encouragement and patience throughout the work.

REFERENCES

[1] Shivam B. Parikh, Pradeep K. Atrey, "Media Rich Fake News Detection: A survey", 2018 IEEE Conference on Multimedia Information Processing and Retrieval.

[2] Akshay Jain, Amey Kasbe, "Fake News Detection", 2018 IEEE International Students' Conference on Electrical, Electronics and Computer Sciences.

[3] Oluwaseun Ajao, Deepayan Bhowmik, Shahrzad Zargari, "Fake News Identification on Twitter with Hybrid CNN and RNN Models"
C3Ri Research Institute Sheffield Hallam University United Kingdom.

[4] B. Markines, C. Cattuto, and F. Menczer, "Social spam detection," in Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web. ACM, 2009, pp. 41–48.

[5] Mykhailo Granik, Volodymyr Mesyura, "Fake News Detection Using Naive Bayes Classifier", 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON).

[6] Shu K., Sliva A., Wang S., Tang J., Liu H., "Fake News Detection on Social Media: A Data Mining Perspective", ACM SIGKDD Explorations Newsletter, 2019, 19(1), 22-36.

[7] Sheryl Mathias, Namrata Jagadeesh, "Detecting Fake News Tweets from Twitter" College of Information Studies, University of Maryland, College Park, USA

[8] Sherry Girgis, Eslam Amer, Mahmoud Gadallah "Deep Learning Algorithms for Detecting Fake News in Online Text" Faculty of Computer Science Modern Academy for Computer Science and Management Technology Cairo, Egypt. IEEE, 2018, Vol.7, No.2.20.

[9] A. Zubiaga, M. Liakata, R. Procter, G. W. S. Hoi, and P. Tolmie, "Analysing how people orient to and spread rumors in social media by looking at conversational threads," *PloS one*, vol. 11, no. 3, p. e0150989, 2016.

[10] V. L. Rubin, N. J. Conroy, and Y. Chen, "Towards news verification: Deception detection methods for news discourse," in Hawaii International Conference on System Sciences, 2015.

[11] V. L. Rubin, Y. Chen, and N. J. Conroy, "Deception detection for news: three types of fakes," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.