FAKE NEWS DETECTION

A PROJECT REPORT

Submitted by

Pranjal Roy (19BAI10008)
Shanzeh Batool (19BAI10017)
Karan Jain (19BAI10095)
Shalini Das (19BAI10139)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Specialization in

Artificial Intelligence and Machine Learning

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRI KALAN, SEHORE
MADHYA PRADESH - 466114

NOV 2020

VIT BHOPAL UNIVERSITY, KOTHRI KALAN, SEHORE MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled **"FAKE NEWS DETECTION"** is the bonafide work of "**Pranjal Roy (19BAI10008), Shanzeh Batool (19BAI10017), Karan Jain (19BAI10095), Shalini Das (19BAI10139)"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here does not form part of any other project / research work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr. S Sountharrajan, Senior Assistant Professor
School of AI & ML division
VIT BHOPAL UNIVERSITY

PROJECT GUIDE
Dr. Ashwin M, Senior Assistant Professor
School of AI & ML division
VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on _______________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First and foremost we would like to thank the Lord Almighty for his presence and immense blessings throughout the project work.

We wish to express our heartfelt gratitude to Dr. S Sountharrajan, Head of the Department, School of Computing Science and Engineering for much of his valuable support and encouragement in carrying out this work.

We would like to thank our internal guide Dr. Ashwin M, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

We would like to thank all the technical and teaching staff of the School of Computing Science and Engineering, who extended directly or indirectly all support.

Last, but not the least, we are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success.

LIST OF ABBREVIATIONS

S.NO	ABBREVIATION	INDICATION
1.	AI	Artificial Intelligence
2.	ANN	Artificial Neural Network
3.	API	Application Programming Interface
4.	BOW	Bags Of Words
5.	CSI	Capture Score Integrate
6.	LSTM	Long -Short Term Memory
7.	ML	Machine Learning
8.	NLP	Natural Language Processing
9.	RNN	Recurrent Neural Network
10.	SVM	Support Vector Machine
11.	US	United States

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The current infodemic of misleading information in everyday access of media outlets such as social media feeds, news blogs, and online newspapers has made it challenging to spot trustworthy news sources, thus increasing the necessity for developing computational tools to provide insights into the reliability of online content. In this paper, we will systematically explain the steps taken to construct a machine learning model for fake news detection. We will go through all the existing algorithms being used in this particular field and their shortcomings. Our proposed system will use recurrent neural networks (RNN) and long-short term memory (LSTM) algorithms. The paper explains in detail the functioning and analysis of each module in our system as well as the implementation of each one. The performance analysis gives the accuracy rate of our model and how it is an improvement over the other existing ones. Finally, we conclude with a few future enhancements which if implemented, can prove to be a blessing in this war against fake news and its exponential spread.

TABLE OF CONTENTS

	References	

INTRODUCTION

1.1 Introduction

Spreading fake news has become a major issue in the media. In this project we have used ML algorithms to build and train models so that it can classify and visualize the news as fake and real with maximum accuracy.

1.2 Motivation for the work

Technology - a boon and a bane. There are a lot of people who rely on newspapers to get a grasp of what's going on overall the world, but as technology progresses, more number of people start to depend on e-materials. But not all the news that users read can be trustable. Fake news can be really harmful to some extent, not only it depreciates the reputation of popular figures, but also can damage one's knowledge about certain topics. It is important for everyone one to know the true facts.

1.3 Introduction to the project

Our project focuses on generating an ML model that will be able to detect the fake news with utmost accuracy. To generate such a model, we would require to train our model using data analysis. We will be using two techniques to develop the model and they are the following:

- Recurrent Neural Network (RNN): It is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs.
- Long Short Term Memory (LSTM): Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies.

1.4 Problem Statement

We can define fake news in the following two ways:

Definition 1 (Broad definition of fake news): Fake news is false news.

Definition 2 (Narrow definition of fake news): Fake news is intentionally false news published by a news outlet.
We consider the above two definitions as our problem statement for this fake news detection project.

1.5 Objective of the work

- Develop a model to detect fake news.
- Use data analysis to train the model.
- Classify the data into fake news and real news with the help of machine learning algorithms, neural networks.

1.6 Organization of the Thesis

This thesis consists of a total 6 chapters, followed by the appendices and references. The current chapter gives the introduction about our thesis. Chapter 2 presents the background of the various algorithms that can be used to develop this model, algorithms that we have used in developing the model and other techniques that have been used to generate this model.
In Chapter 3, we discuss the disadvantages and the limitations of the existing models, and what our model states, followed by summary. Chapter 4 presents a detailed explanation of design and implementation of the ML model and Chapter 5 gives a pictorial and graphical representation of the performance of the model. In Chapter 6, we discuss how we enhance the existing model followed by conclusion.

1.7 Summary

With the growth of technology, spreading of fake news on online platforms has become a common problem these days. In this project we have decided to overcome this problem with the help of ML algorithms to train our model. We have chosen the LSTM algorithm to train our model that is an extension of RNN and works very well with big data giving high accuracy. This thesis contains all the necessary details and concepts that have been used in the fake news detection project.

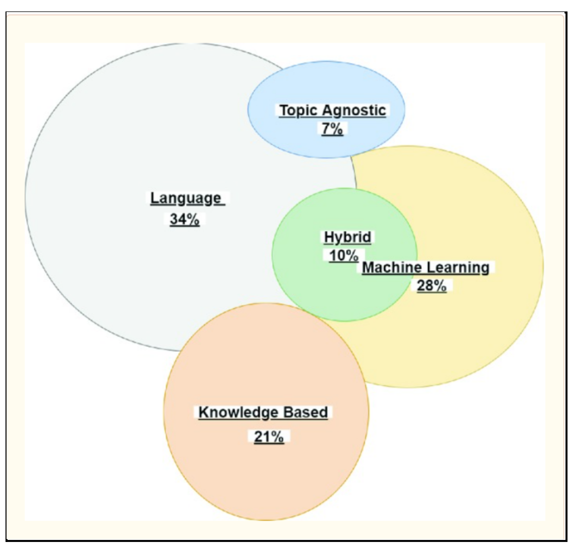
LITERATURE SURVEY

2.1 Introduction

Fake news is not a new concept. Before the era of digital technology, it was spread mainly through newspapers or on a one-to-one basis, but now social media makes the job much easier. A variety of approaches exist to spot fake news. By conducting a scientific literature review, we identify and categorize the currently available and the most commonly used methods to spot fake news and the way these approaches are often applied in several situations.

2.2 Core Area of the Project

The following categories of approaches for fake news detection are proposed -



[Figure 2.2. Categories of fake news detection approaches as a result of systematic literature review]

Language approach focuses on the utilization of linguistics by a person or a software program to detect fake news. Most of the people liable for the spread of faux news have control over what their story is about, but they will often be exposed through the design of their language (Yang et al. 2018). This approach considers all the words during a sentence and letters during a word, how they're structured and the way it fits together during a paragraph (Burkhardt 2017). The main target is therefore on grammar and syntax of the data.(Burkhardt 2017). It can further be classified into Bag of Words (BOW) (Burkhardt 2017), Semantic Analysis (Chen et al. 2017b) and Deep Syntax (Stahl 2018).

Topic-agnostic approach detects fake news by not considering the content of articles but rather identifying the topic-agnostic features. The approach uses linguistic features and web mark-up capabilities to spot fake news (Castelo et al. 2019). Some samples of topic-agnostic features are 1) an outsized number of advertisements, 2) longer headlines with eye-catching phrases, 3) different text patterns from mainstream news to induce emotive responses 4) presence of an author name (Castelo et al. 2019; Horne and Adali 2017).

Machine learning approach uses machine learning algorithms to identify fake news. This is often achieved by using differing types of training datasets to refine the algorithms. Datasets are used to train the algorithms to spot fake news. One of the methods to create these datasets is through crowdsourcing.

Perez-Rosas et al. (2018) created a fake news data set by first collecting legitimate information on six different categories like sports, business, entertainment, politics, technology and education (Pérez-Rosas et al. 2018).

A machine learning approach called the rumor identification framework has been developed that legitimizes signals of ambiguous posts in order to alert the people of posts which may be fake (Sivasangari et al. 2018).The framework is made to combat fake tweets on Twitter and focuses on four main areas; the metadata of tweets, the source of the tweet; the date and area of the tweet, where and when the tweet was developed (Sivasangari et al. 2018). Supporting this framework, the spread of gossip is collected to make datasets with the utilization of a Twitter Streaming API (Sivasangari et al. 2018). Twitter has developed a possible solution to spot and stop the spread of misleading information through fake accounts, likes and comments (Atodiresei et al. 2018) - the Twitter crawler, a machine learning approach works by collecting tweets and adding them to a database, making comparison between different tweets possible.

Knowledge-Based approach aims at using sources that are external to verify if the news is fake or real and to spot the news before the spread thereof becomes quicker. The challenging problem with a number of these fact checking methods is the speed at which fake news spreads on social media. Microblogging platforms like Twitter causes small pieces of false information to spread very quickly to an outsized number of individuals (Qazvinian et al. 2011). There are three main categories; (1) Expert Oriented Fact Checking, (2) Computational Oriented Fact Checking, (3) Crowdsourcing Oriented Fact Checking (Ahmed et al. 2019).

Hybrid approach proposes a hybrid model which helps to spot fake news on social media through employing a combination of human and machine learning to assist identify fake news (Okoro et al. 2018). There are three generally prescribed elements of faux news articles, the primary element is the text of a piece of writing , the second element is the response that the articles receive and lastly the source used that motivates the news story (Ruchansky et al. 2017). A hybrid model called CSI (capture, score, integrate) has been developed and functions on the elements : (1) capture - the method of extracting representations of articles by employing a Recurrent Neural Network (RNN), (2) Score – to make a score and representation vector, (3) Integrate – to integrate the outputs of the capture and score leading to a vector which is employed for classification (Ruchansky et al. 2017).

2.3 Existing Algorithms

2.3.1 Passive Aggressive

Passive Aggressive algorithms are online learning algorithms. Such an algorithm remains passive for a correct classification outcome, and turns aggressive in the event of a miscalculation, updating and adjusting. Unlike most other algorithms, it does not converge. Its purpose is to make updates that correct the loss, causing very little change in the norm of the weight vector.

2.3.2 Naive Bayes

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

2.3.3 Support Vector Machine

A Support Vector Machine (SVM) is a monitored machine learning model using two-group classification algorithms. In this we need to first identify the right hyperplane, incase of multiple planes. Then we need to classify the two classes and finally place the correct hyperplane. It works really well in clear margins and is also effective in high dimension space. But it doesn't perform well in case of large datasets.

2.3.4 Recurrent Neural Networks

Recurrent neural networks, also referred to as RNNs, are a category of artificial neural networks that allow previous outputs to be used as inputs while having hidden states. They have an advantage over the vanilla neural networks as they include a memory effect. But they can be further improved by using LSTM networks which will solve the vanishing gradient problem.

2.3.5 Long-Short Term Memory

LSTM is an extension of RNN architecture used in the field of deep learning. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.

2.4 Research Issues/Observations from Literature Survey

While observing the above machine learning models, the analysis and comparison of these shows that Naive Bayes algorithm has the minimum accuracy of 71.84% predicting a news correctly. Then comes the Support Vector Machine and Recurrent Neural Network algorithms with accuracies of 87.37% and 90.47% respectively to predict the news correctly. Then we observe the most accurate algorithm namely the Long-Short Term Memory algorithm which predicts the news with 99.70% accuracy.

2.5 Summary

To summarise, the core area of the project includes a variety of approaches, majorly language approach, topic-agnostic approach, machine learning approach, knowledge based approach and the hybrid approach. We have broadly five different types of algorithms to build a fake news detection machine learning model namely :-

- Passive Aggressive
- Naive Bayes
- Support Vector Machine
- Recurrent Neural Networks
- Long-Short Term Memory

SYSTEM ANALYSIS

3.1 Introduction

In this chapter, we discuss disadvantages and the limitations of already existing systems, what we propose in our model and how our model makes a difference. This is followed by a brief summary of this chapter.

3.2 Disadvantages /Limitations in the Existing System

Some already existing apps and algorithms - and their limitations:

1. Test news (app): Only functions when a link is provided to it. Otherwise it is invalid for almost every news article.
2. Fake finder (app): An AI fake news detection and filter that is developed by twitter and still being worked on. This AI however is centered only on one social media i.e twitter.
3. Newscope (app): Was a famous app but was not completely AI based. Now due to their lack of funds they have disabled expert reviews and the app has lost its accuracy and become biased.
4. Passive aggressive classifier: If the outcome is correct, it remains passive but in case of miscalculation it becomes aggressive, updating and adjusting. The accuracy rate varies depending on the size of the dataset.
5. Naive Bayes: It accepts every feature as independent, but it is almost impossible to get a set of predictors which are completely independent. It has the least accuracy rate of about 74.81%.
6. Support Vector Machine: Though, it has a good accuracy rate, but the data transforms are complex and the resulting graphs are very difficult to interpret.

3.3 Proposed System

In our proposed system, we construct a model to classify the data into fake and real news with the help of recurrent neural networks (RNN) and long-short term memory (LSTM) algorithms. We will first import and install various datasets and libraries . Then we'll assign values 0 and 1 to fake and true news respectively to create a table. After this cleaning of the original data will be done by removing the stop-words and counting unique words. Here, we can visualize the cleaned data. Now in order to train our machine learning model, we will first divide our dataset into training and testing data. In order to feed this data into the ML model we perform tokenization and padding on it. Next, with the help of LSTM networks as an extension of the neural networks, we build an LSTM model. Using the embedded layer, we can reduce the usage of variables significantly. Now we fit our model to the training data and train it. The fake news detection model is now ready to predict and classify news as fake or true and also the accuracy of the predictions.

3.4 Summary

Our proposed system aims at reliably predicting and classifying the given piece of news as fake or true with the help of recurrent neural networks (RNNs) and long-short term memory (LSTM) algorithms. With the help of our proposed machine learning model, we lay a solid foundation for efficient detection of fake news by overcoming the limitations of the existing systems and trying to achieve higher accuracy rates.

SYSTEM DESIGN AND IMPLEMENTATION

4.1 Introduction

To build a machine learning model for fake news detection, a step by step procedure is to be followed which are given in the form of modules below.

4.2 Understand The Problem And Business Case

Right now, we live in a world of mis-information and fake news. The goal of this hands-on project is to detect fake news based on Recurrent Neural Networks. Natural language processing (NLP) works by converting word (text) into numbers. These numbers are then used to train an Al/ML model to make predictions. Al/ML-based fake news detector is crucial for companies and media to automatically predict whether circulating news is fake or not. In this case study, we will analyze thousands of news texts to detect if it is fake or not.

4.3 Import Libraries And Datasets

The task involves importing libraries from packages and then importing the real and the fake news datasets.

4.4 Perform Exploratory Data Analysis

This task focuses on adding a target class column in the dataset which will categorise the news as real or fake. Then comes the concatenation of the two real and fake datasets. Finally combining the Title and Text datasets and exploring it.

4.5 Performing Data cleaning

This task involves cleaning of data, in the more obvious sense removing stopwords.
Stopwords are the words which are commonly used, and don't add much meaning to the sentence. To do this, we first need to download the stopwords library from ‘nltk’, and in addition to the library we add more common words to narrow the filtration. We make a

function to remove the stopwords. As we observe that most stopwords are less than 3 letters, we apply this criteria also. Then we perform this function to the dataframe and obtain clean data or data without stopwords. Now we are left with words that have meaning but are not connected. So, we use this opportunity to count the total unique words, which will be used in the next task in order to visualise the data.

4.6 Visualising cleaned up data

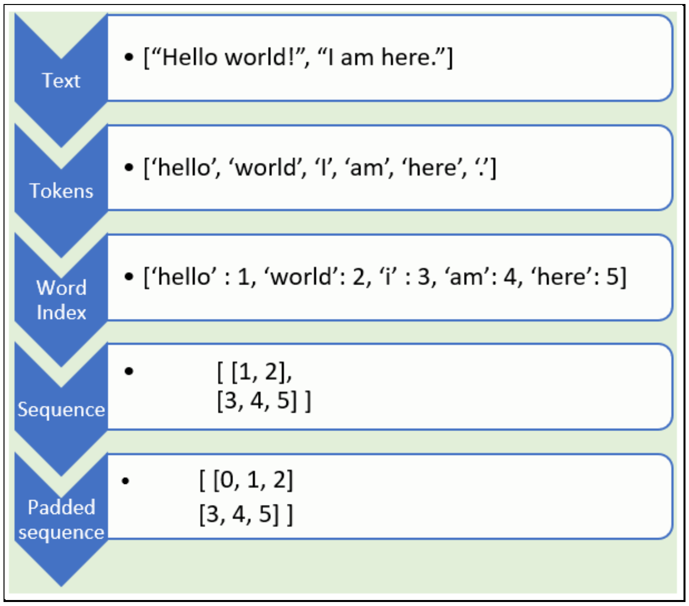
In this task, we will mainly focus on how often the word is used. We use word cloud for visualisation. Word cloud is a powerful tool for visualization of text data. So, we plot the word cloud for fake data as well as real data. Using this tool, we can compare the two visualization. Maximum length of words will be required to create word embeddings. So, we obtain the maximum length of the words using nltk.word_tokenize which divides each of the words to get the unique words. We can also visualize the distribution of number of words in the data.

4.7 Prepare the Data by Performing Tokenization and Padding

In the previous task we did data visualization. In order to prepare the data before training the LSTM network, we need to take the data and divide it into training and testing data. The idea is to train the machine learning model on the training data and once it is trained, to see how it is performing on the testing data which is essentially the data that the model has never seen before during training. After this we do padding and now we are ready to train the model.

Tokenization - Text information can’t be fed into the machine learning model as computers learn the news in the form of numbers. So we need to convert the text or encode the sentences into tokens or a sequence of integers and train the AI-ML model using these. We feed the training data into the ‘fit-on-texts’ method and it returns the tokenizer. This tokenizer is essentially used to tokenize the data.

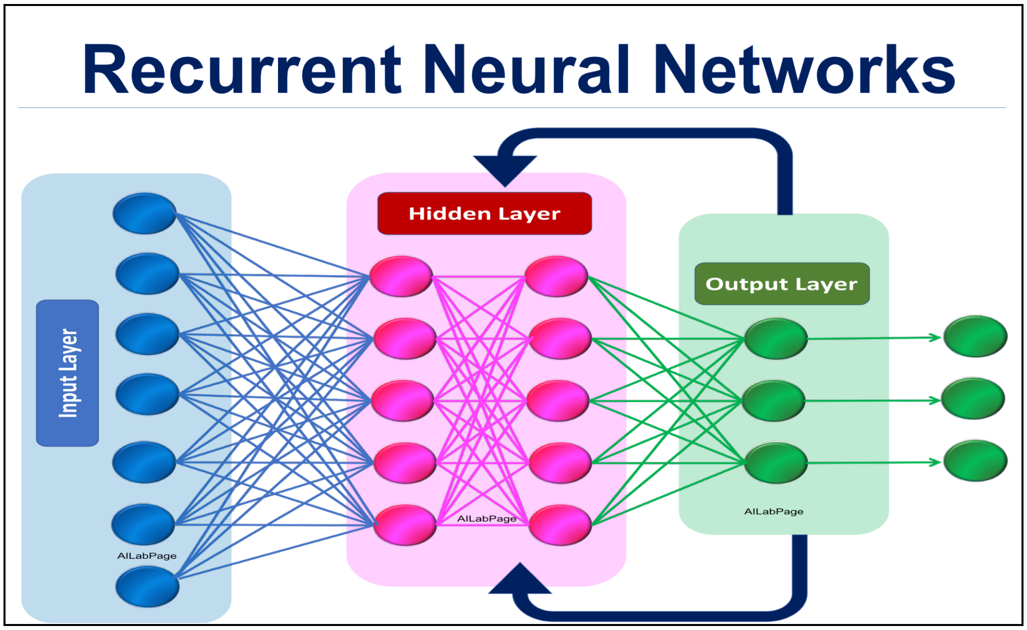
Padding - We apply padding to the training data as well as the testing data by specifying the maximum length of the document. Zeros are added to ensure that all the samples have the same length.



[Figure 4.7. Tokenization & Padding]

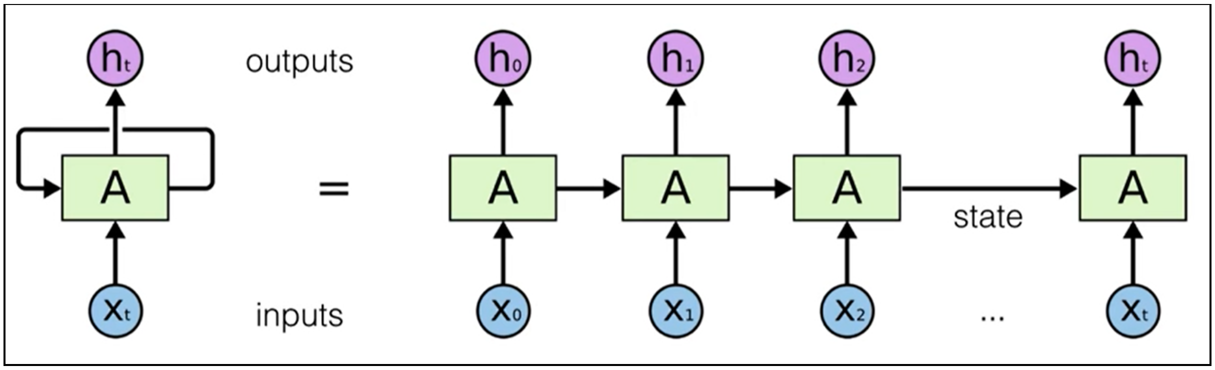
4.8 Understand the Intuition behind Recurrent Neural Networks (RNN)

In the previous task we did some pre-processing on the data. Feedforward neural networks are vanilla networks in which fixed size inputs are directly mapped to fixed size outputs. All neurons here are connected to all neurons in the subsequent layers and the data propagates from the left-hand-side to the right-hand-side. Like a linear equation, we feed in the independent variables and we get the dependent variable. It is a one-to-one mapping without any dynamics. In these basic neural networks, the drawback is that we don't remember what happened in the past. We have no time dependency or memory effect.



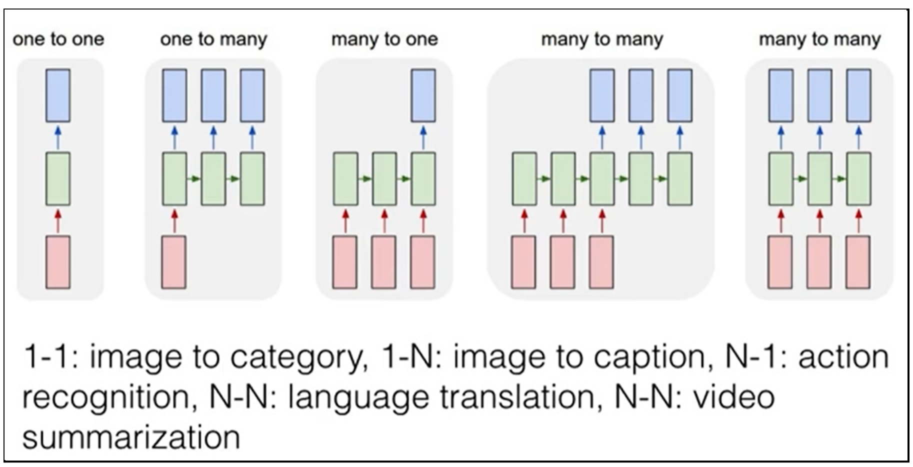
[Figure 4.8.1. Recurrent Neural Networks (RNNs)]

Whereas recurrent neural networks or RNNs are a type of artificial neural networks in which the output is fed back as the input, so the output not only depends on the input, it also depends on what happened in the past. Hence we observe a **memory effect** as we have added an extra dimension of time with the help of the feedback loop. What happens at time ‘t’, will depend on the input as well as the previous time stamp ‘t-1’ and this works great with a sequence of text.



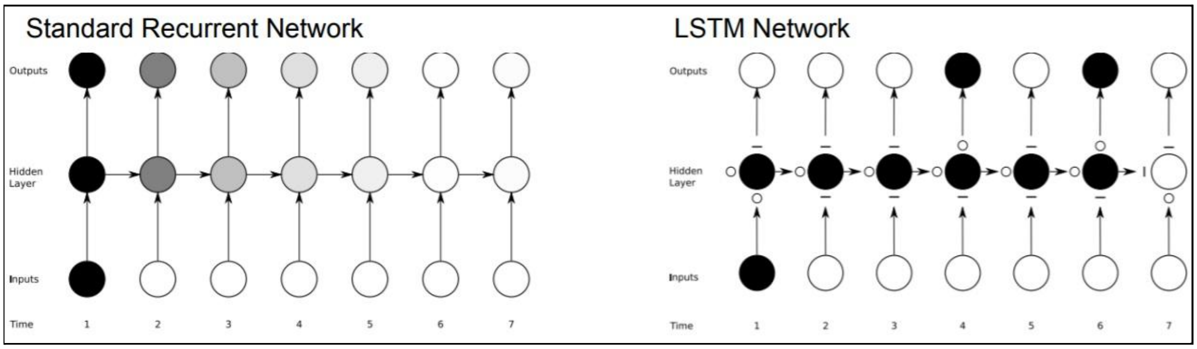
[Figure 4.8.2. Memory Effect in RNN]

Feedforward neural networks are very rigid because of the fixed number of inputs and outputs. Recurrent neural networks offer a huge advantage over feedforward ANNs as they allow us to work with a sequence of inputs or outputs or both. RNNs may have one-to-one mapping used in regression and classification, many-to-many mapping with a sequence of inputs and outputs used in language translation, one-to-many mapping used in image captioning or many-to-one mapping used in sentiment analysis.



[Figure 4.8.3. Learning Sequences]

Gradient descent is an optimisation algorithm used to balance the weights in an ANN. We try to formulate the cost function which we need to minimize. So we calculate the gradient by moving in the negative direction and stop when we get optimized values of the weights. A challenge with training the RNN is the **Vanishing Gradient Problem**. In RNN, the network generates predictions based on the labels. The error signal taken from this is used to update the weights. Supposing the gradient is 0.1 and if we apply chain rule and keep on multiplying 0.1 with 0.1 and so on, we observe a phenomenon called the vanishing gradient problem. In this, the gradient becomes too small and we have to stop training the weight as they can't be updated anymore. This problem is overcome by using **Long Short Term Memory or LSTM**.



[Figure 4.8.4. Vanishing Gradient Problem in RNN]

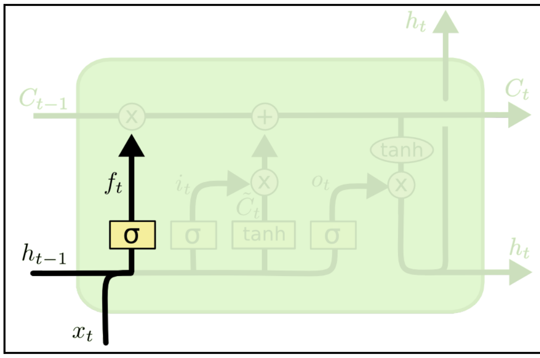
4.9 Understand the intuition between LSTM networks

In practice RNN has failed to establish long term dependencies but LSTM networks are a type of RNN that remember long term dependencies by default. To understand this better we can take an example of tree color and weather forecast.

If the tree is evergreen then the statement "The tree is green in colour" will be true forever. Now let us look at the weather of a particular place and the statement is, "I live in Northern Canada and it's winter over there." This statement is true only when it is winter in Northern Canada that will change over a period of time. So here depending on the time the statement can either be true or false. This is where LSTM can work better on timelines.

The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates. The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.

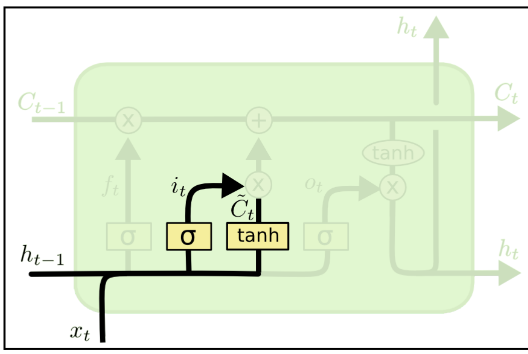
The first step in our LSTM is to decide what information we're going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer." It looks at h_{t-1} and x_t, and outputs a number between 0 and 1 for each number in the cell state C_{t-1}. A 1 represents "completely keep this" while a 0 represents "completely get rid of this."



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \;+\; b_f\right)$$

[Fig. 4.9.1. The Sigmoid Layer]

The next step is to decide what new information we're going to store in the cell state. This has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update. Next, a tanh layer creates a vector of new candidate values, $\tilde{C}_t$, that could be added to the state.

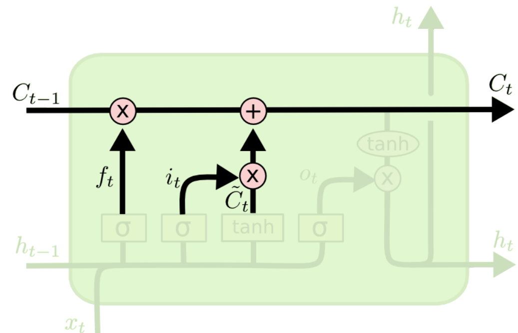


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \;+\; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \;+\; b_C)$$

[Fig. 4.9.2. tanh Layer]

It's now time to update the old cell state, C_{t-1}, into the new cell state C_t. The previous steps already decided what to do, we just need to actually do it.

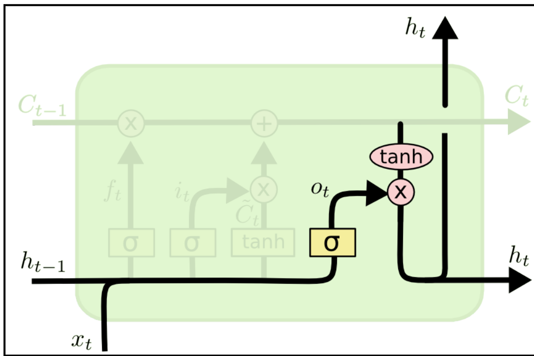
We multiply the old state by f_t, forgetting the things we decided to forget earlier. Then we add $i_t * \tilde{C}_t$. This is the new candidate value, scaled by how much we decided to update each state value.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

[Fig. 4.9.3. Get the new candidate value]

Finally, we need to decide what we're going to output. This output will be based on our cell state, but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through $\tanh$(to push the values to be between $-1$and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

[Fig. 4.9.4. Get the output through sigmoid gate]

4.10 Build and Train the LSTM model

We first use an embedding layer to build the model. Embedding layers learn the low-dimensional continuous representation of input discrete variables.

For example, if we have 100000 unique values in our data then with the help of an embedding layer the model can learn a way to represent them in 200 variables. This in-turn helps subsequent layers learn more effectively with less compute resources.

We also divide the data into training data and testing data. Testing is the subset of data that the model has never seen before during training and that will happen after the model is trained.

When we train the model we set the number of epochs to 2. This reduces the error while training the model with the help of our samples.

4.11 Assess Trained LSTM Performance

Finally the model makes it's prediction. We import the accuracy_score function with the help of sklearn.metrics library to calculate the accuracy. We also import the confusion_matrix to visualize the classification and misclassification of data.

4.12 Summary

While designing the system and implementing the code we have a sequence of steps that we need to follow. First we start with understanding the problem and business care as we need to have a clear idea on what solution we are going to develop for that problem. Second, we import the necessary data and libraries that we need during the implementation of code. Then we perform an exploratory data analysis that focuses on adding a target class column in the dataset which will categorise the news as real or fake. Next we perform data cleaning to remove the stopwords i.e. the less important words. Then we focus on visualizing the cleaned up data with the help of word cloud. In order to prepare the data before training the LSTM network, we need to take the data and divide it into training and testing data. We also prepare the data by performing tokenization and padding. Next we try to understand the intuition between RNN and LSTM and how the vanishing gradient problem of RNN is overcomed by LSTM. After this we are ready to build and train the LSTM model. Finally we assess the trained LSTM model performance with the help of testing data and calculate its accuracy.

PERFORMANCE ANALYSIS

5.1 Introduction

Since the model is being trained with the help of an ML algorithm i.e. LSTM we need to take into account the performance of our model when we do testing.

5.2 Performance Measures

The table below shows the accuracy rates of different algorithms:

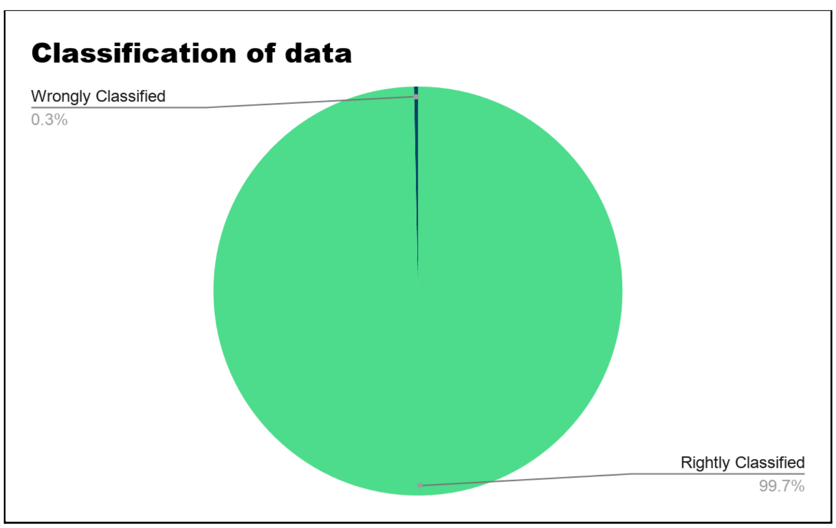
Models	**Accuracy rates**
Naive Bayes	71.84%
SVM	87.37%
Neural Network	90.47%
LSTM	99.688%

[Table 5.2. Various Models vs. Accuracy Rates]

We can see from table 5.2 that LSTM has the highest accuracy compared to other algorithms.

5.3 Performance Analysis

When the performance of this model is analyzed we found that some of the data were misclassified as our model's accuracy was 99.688%. The following pie chart shows the rightly classified and wrongly classified data while testing our LSTM model:



[Figure 5.3. Performance Analysis]

Below we have the countplot of isfake vs. count. In the y-axis, 0 represents fake while 1 represents true. On the x-axis we get the count of words. We plot the countplot graph for fake and real news with the help of plotly:

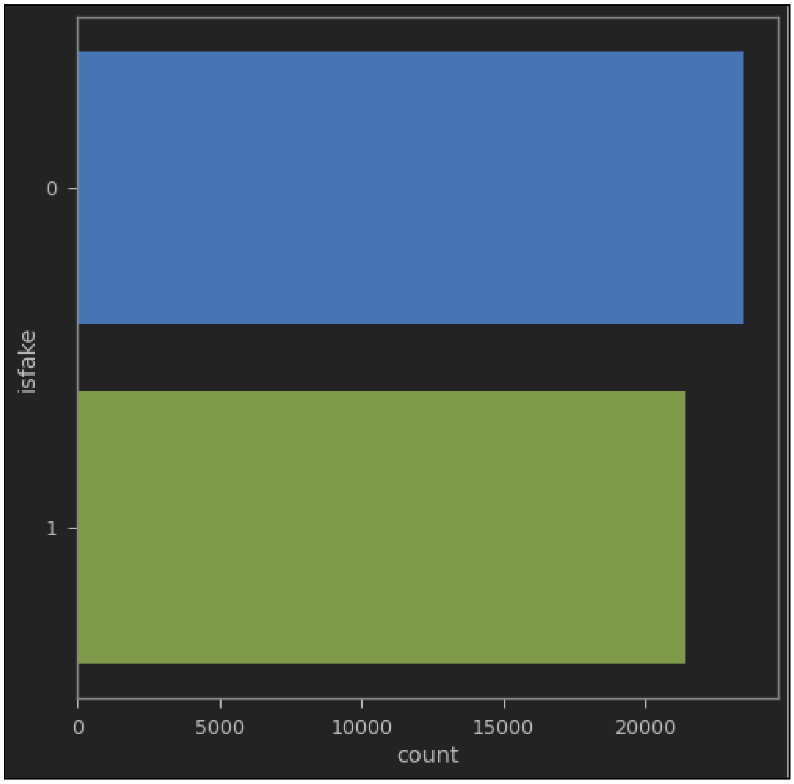


Fig. 5.3.2

5.4 Summary

From the above table, graph and pie chart we can see that LSTM has been able to successfully classify the data with a higher accuracy(99.688%) as compared to other algorithms. This performance can definitely be further increased.

FUTURE ENHANCEMENT AND CONCLUSION

6.1 Introduction

Future enhancement is improvement of something which is existing in the present, and here it is the improvement of the Fake News Detection ML model which is discussed above. The model although has more than 99% of accuracy, but to implement this model to be useful in the daily real life situations, it is necessary to integrate this with an app or a website so that common people can access it easily.

6.2 Limitation/Constraints of the System

This model currently works on datasets which are a part of the US newspapers and articles. Therefore the detection of the fake news is only limited to the news related to the US and its affairs. This is a limitation for users living in other countries where fake news is prominent amongst social media platforms and even in news articles and online news displaying platforms.

This model is also not accessible by common people as not everyone has the knowledge and technology to use Machine Learning models. Also people want a simple and clean interface which works as a link between the model and the user using it.

This model does give us 99.7% accuracy, but this model is not much time efficient and its progress is saved remotely and not in any cloud service. So each time this model is run on a new device it takes a lot of time to run, train and test.

The model also takes only text as an input which is not possible sometimes by a user to type as a whole.

6.3 Future Enhancements

Some enhancements that can be made to make this model more usable, accessible and efficient to use in daily life:-

- The model can be integrated with a responsive website or an ios/android application which interacts with the user and gives the output.
- The application or website can be designed with Google's ML API kit so that it can retain and can take text from images which are provided by a user as an input.

- Also the progress of the model can be saved in a cloud server/platform so that it can become time efficient for it to give output as an input is given.
- The model can also be trained with datasets of different regions of the world with different languages so that it can become more user friendly.
- This model can also be integrated with social media applications where fake news and rumors have become prominent.

6.4 Conclusion

The improvements and future enhancements are therefore much necessary for the model to increase its usability, accessibility and efficiency.

REFERENCES

1. Dylan de Beer and Machdel Matthee : *Approaches to Identify Fake News: A Systematic Literature Review,* 2020

2. Jiawei Zhang, Bowen Dong, Philip S. Yu *: FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network, 2019*

3. Kelly Stahl., "Fake news detection in social media", B.S. Candida

4. Perspective", ACM SIGKDD Explorations Newsletter, 2017, 19(1), 22-36.

5. Rubin, V., Conroy N., Chen Y., Cornwell S., "Fake News or Truth? Using Satirical Cues to Detect

6. Shu K., Sliva A., Wang S., Tang J., Liu H., "Fake News Detection on Social Media: A Data Mining

7. Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, Rada Mihalcea : *Automatic Detection of Fake News,* 2017

8. https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

9. https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

10. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

11. http://theprofessionalspoint.blogspot.com/2019/03/advantages-and-disadvantages-of-naive.html