

# COMPARATIVE ANALYSIS OF VARIOUS ALGORITHMS FOR FAKE NEWS DETECTION

*A thesis submitted in partial fulfillment  
of the requirements for the degree of*

**Bachelor of Technology**  
In  
**Computer Science and Engineering**

by

**Prithwiraj Samanta**

*Under the guidance of*

**Dr. Rashmi Panda**



**भारतीय सूचना प्रौद्योगिकी संस्थान राँची**

**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, RANCHI**

(An Institution of National importance under an act of Parliament)

(Ranchi - 834010), Jharkhand

**Department of Computer Science and Engineering**



# Indian Institute of Information Technology Ranchi

---

## Certificate

This is to certify that the thesis entitled "**Comparative Analysis of various Algorithms for Fake New Detection**" is a Bonafede record of work carried out by **Prithwiraj Samanta**, under my supervision and guidance, for the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology (Honours)** in Computer Science and Engineering at the Indian Institute of Information Technology, Ranchi. The thesis has fulfilled all the requirements as per the regulations of the institute and in my opinion, reached the standard for submission.

---

**Dr. Rashmi Panda**

Assistant Professor

Indian Institute of Information Technology

**Date: December 2022**

**Place: IIIT Ranchi**

# Acknowledgment

I am greatly indebted to my project supervisor Prof. Rashmi Panda for her invaluable guidance and encouragement throughout the course of this project. I would like to take this opportunity to express my sincere and profound gratitude to her.

I would also like to gratefully acknowledge the suggestions and discussions received from my teammate Satya.

My sincere thanks also go to all the faculty of CSE for their kind co-operation in all spheres during my project work. I would also like to thank all others whose direct or indirect help has benefited me during my stay at IIIT Ranchi.

Furthermore, I would like to thank all other friends and fellow students of IIIT Ranchi and particularly my CSE mates for sparing a great time throughout my B. Tech and very good support I received from them during my stay.

Last but not the least, I thank my parents and all my family members for the support and motivation they have provided throughout my project.

**Prithwiraj Samanta**

# Abstract

The advent of the World Wide Web and the rapid adoption of social media platforms (such as Facebook and Twitter) paved the way for information dissemination that has never been witnessed in human history before. With the current usage of social media platforms, consumers are creating and sharing more information than ever before, some of which are misleading with no relevance to reality. Automated classification of a text article as misinformation or disinformation is a challenging task. Even an expert in a particular domain has to explore multiple aspects before giving a verdict on the truthfulness of an article. In this work, we have created an analysis report of various algorithms (particularly LSTM + CNN + Attention + Transformer Models) for the automated classification of news articles. Our study explores different textual properties that can be used to distinguish fake contents from real. By using those properties, we train a combination of different machine learning algorithms using various methods and evaluate their performance on real-world datasets.

# Content

<b>Acknowledgment</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>iv</b>
<b>1 Introduction</b>	<b>6</b>
<b>2 Review of Literature</b>	<b>8</b>
<b>3 Implementation</b>	<b>10</b>
3.1 Data Pre-processing .....	10
3.2 LSTM + CNN Models .....	11
3.3 RNN + Attention Models .....	15
3.4 Transformer Models .....	16
<b>4 Results and Discussion</b>	<b>18</b>
4.1 Result .....	18
4.2 Discussion .....	18
<b>5 Conclusion and Future Work</b>	<b>20</b>
5.1 Conclusion .....	20
5.2 Future Work .....	20
<b>List of Abbreviation</b>	<b>vi</b>
<b>References</b>	<b>vii</b>

# List of Figures

3.1	Graph of Loss of Model-1 .....	12
3.2	Graph of Accuracy of Model-1 .....	12
3.3	Graph of Loss of Model-2 .....	13
3.4	Graph of Accuracy of Model-2 .....	13
3.5	Graph of Loss of Model-3 .....	13
3.6	Graph of Accuracy of Model-3 .....	13
3.7	Graph of Loss of Model-4 .....	14
3.8	Graph of Accuracy of Model-4 .....	14
3.9	Graph of Loss of Model-5 .....	14
3.10	Graph of Accuracy of Model-5 .....	14
3.11	RNN + Attention Architecture .....	15
3.12	Graph of Loss of Model-6 .....	16
3.13	Graph of Accuracy of Model-6 .....	16
3.14	Transformer Architecture .....	17
3.15	Graph of Loss of Model-7 .....	17
3.16	Graph of Accuracy of Model-7 .....	17

# List of Tables

4.1	Summary of Result of all models .....	10
-----	---------------------------------------	----

# 1

## Introduction

With the advancement of technology, digital news is more widely exposed to users globally and contributes to the increment of spreading hoaxes and disinformation online. Fake news can be found through popular platforms such as social media and the Internet. There have been multiple solutions and efforts in the detection of fake news where it even works with artificial intelligence tools. However, fake news intends to convince the reader to believe false information which deems these articles difficult to perceive. The rate of producing digital news is large and quick, running daily at every second, thus it is challenging for machine learning to effectively detect fake news.

In the discourse of not being able to detect fake news, the world would no longer hold value in truth. Fake news paves the way for deceiving others and promoting ideologies. These people who produce the wrong information benefit by earning money with the number of interactions on their publications. Spreading disinformation holds various intentions, in particular, to gain favor in political elections, for business and products, done out of spite or revenge. Humans can be gullible and fake news is challenging to differentiate from normal news. Most are easily influenced especially by the sharing of friends and family due to relations and trust. We tend to base our emotions on the news, which makes accepting not difficult when it is relevant and stance from our own beliefs. Therefore, we become satisfied with what we want to hear and fall into these traps.

**Problem Statement:** In this report, we propose to study the fake news detection (including the articles and subjects) problem in the digital era. Based on various types of articles consisting of fake and real heterogeneous information from online websites both textual contents/profile/descriptions, we aim at identifying fake news from online. We formulate the fake news detection problem as a credibility inference problem, where the real ones will have higher credibility while unauthentic ones will have a lower one instead.

The fake news detection problem is not easy to address due to the following reasons:

- i. **Problem Formulation:** The fake news detection problem studied in this report is

a relatively new research problem, and a formal definition and formulation of the problem are required and necessary before studying the problem.

- ii. Textual Information Usage: For the news articles, creators, and subjects, a set of textual information about their contents, profiles, and descriptions can be collected from online social media. To capture signals revealing their credibility, an effective feature extraction and learning model will be needed.
- iii. Heterogeneous Information Fusion: In addition, as mentioned before, the credibility labels of news articles, creators, and subjects have very strong correlations, which can be indicated by the authorship and article-subject relationships between them. Effective incorporation of such correlations in the framework learning will be helpful for more precise credibility inference results of fake news.

This report proposes a different methodology to create models which will detect if an article is authentic or fake based on its words, phrases, sources, and titles, by applying supervised machine learning algorithms on an annotated (labeled) dataset, that are manually classified and guaranteed. Then, feature selection methods are applied to experiment and choose the best-fit features to obtain the highest precision, according to confusion matrix results. We propose to create the model using different classification algorithms. The product model will test the unseen data, the results will be plotted, and accordingly, the product will be a model that detects and classifies fake articles and can be used and integrated with any system for future use.



# 2

## Review of Literature

The radical change in the availability of technology has brought has been a great contribution to every aspect of life. Contrary to that, as fresh news content is rapidly being generated it is as important to test the truthfulness of the content and credibility of the source. Fake news Detection has been around actively for a decade now but the rate of producing digital news is large and quick, running daily at every second, thus it is challenging for machine learning to effectively detect fake news and it is very important to keep developing the techniques to keep up with the pace.

This paper[1] is a perfect way to dive into the vast spectrum of Fake News Detection. Starting from the introduction and definition of fake news detection, it takes a very good look at various methods implemented to identify fake news and prevent it from releasing publicly. It talks about different machine learning traditional models used and ongoing deep learning models being used to improve the accuracy of identifying fake news.

The paper[2] uses a Fake news challenge dataset which has 75000 instances is divided into train and validation data. The baseline models used are CNN models and BERT. The proposed model uses an attention layer with CNN and RNN. To improve the accuracy dropout layer and max-pooling are used. The best accuracy it achieved is 71.21% on the competition test set.

Using Transformer Network[3] The dataset used is a combination of three datasets which are the WILD dataset, LIAR dataset, and a dataset taken from one of the Kaggle competitions. Models like fine-tuned BERT, CNN-LSTM, CNN are used with the embedding layer. The best accuracy of 90% is achieved by BERT.

Recurrent Neural Network[4] is proficient at detecting patterns on Sequential data. This paper experiments with various models containing the RNN layer. The dataset used consists of 5800 tweets. The model with the LSTM layer performs the best with an accuracy of 82.29%.

This paper[5] tells us about the accuracy of different models i.e., Naive Bayes, Decision trees, SVM, Neural Networks, Random Forest, XG Boost on different datasets. It shows that Random Forest yielded 65.6 accuracies on liar dataset [liar dataset] preceding naive Bayes, SVM, logistic regression, and decision tree with 63.7, 63, 62.5, 60 accuracies.

Xuzhou[6] 2015 proposed a rumor detector that identifies trending rumors on Twitter. The detector, which searches for rare but informative phrases, combined with clustering and a classifier on the clusters, yields surprisingly good performance. According to this detector, on Twitter out of 50 candidate statements, about one-third of them are real rumors.

A paper published by Ahmed[7] gives us all the relevant information regarding the implementation of machine learning on fake news. It gives us an overview of the Methodology, pre-processing, and implementation tasks of a model. The passive-aggressive classifier gives 0.93 accuracies on the fake news dataset which is the maximum of all the classifiers used.

A hybrid deep learning model[8] (a combination of CNN and LSTM) was proposed which experiments with dimension reduction techniques and pre-processing. The dimensionality reduction methods it uses are principal component analysis [2018 on using Principal Component Analysis] and Chi-square [chi square reduction] on the fake news challenge dataset. The best accuracy is yielded by CNN-LSTM with PCA, which is 97.8%.

A study[9] shows an implementation of the deep two-path semi-supervised learning model “DSTL” on the PHEME dataset. To train and test the model both labeled and unlabelled dataset is used. The model contains three CNNs. The performance of the DSTL is inspected with different ratios of labeled data. The proposed model surpasses the F-score of bidirectional recurrent neural (BRNN), 35.85%, by yielding an F-Score of 57.98% with 30% labeled data.

A paper[10] published in 2020 proposes a tree structure model having two branches with the idea of using CNN and LSTM parallel. As we know CNN is good at extracting spatial features and LSTM is good at finding long-term dependencies in data. Further, it suggests concatenating both output vectors and implementing the SoftMax layer. The input data of the branches differ due to pre-processing methods and corpus representation. CNN yields the best accuracy when used with character level embedding where content like URL information, emoji, stop words are also taken into account. RNN variants like LSTM, Bi-LSTM, GRU input data are ready after pre-processing and applying word embedding methods like FastText, which can embed the words successfully which are not present in the corpus. It yields an F1-score of 0.89 on a self-mined dataset which consists of 17,289 Turkish tweets.

# 3

## Implementation

### 3.1 Data Pre-processing

The text data which is available to us for use in fake news detection is full of noisy information and present in a format that can't be directly used by our algorithms. So, we have done the data pre-processing to transform the data into a useable form.

We started with removing the null values present in the dataset. We have dropped the entire row in which any column is null.

We have expanded the contractions(short forms and shorthands) used in the English text using the contractions library of python. We converted the text into lowercase and have split the text into a word list.

All forms of punctuation and stop words have been removed from the processed word list including special characters and numbers. The removal of all non-English words from the word list is performed and then again converted back to text.

Till now most of the noise and unwanted data are removed. Now we will be formatting the text to be used by models. We have to use a TF-IDF vectorizer to extract the features from the corpus(processed text) and convert them to a word vector.

We have also created a separate form of a dataset using tokenizer instead of vectorizer. At last, we save both formats using the pickle library of python. We have generated both the format used by common ML algorithms. We tested the performance of our datasets using Multinomial Naïve Bayes Classifiers.

We selected one of the datasets among 3 based on the performance. Now we are ready to proceed with preparing the selected dataset for Deep Learning.

For preparing the data for Deep Learning, we will be using stemming instead of vectorization or tokenization. We will be using a python library, Port Stemmer. We used a one-hot encoder to encode the words and then generate the embedding matrix.

We convert the matrix into a NumPy array before feeding it to the network.

## 3.2 LSTM + CNN Models

In theory, classic RNNs can keep track of arbitrary long-term dependencies in the input sequences. The problem with vanilla RNNs is computational (or practical) in nature: when training a vanilla RNN using back-propagation, the long-term gradients which are back-propagated can "vanish" (that is, they can tend to zero) or "explode" (that is, they can tend to infinity), because of the computations involved in the process, which use finite-precision numbers. RNNs using LSTM units partially solve the vanishing gradient problem, because LSTM units allow gradients to also flow unchanged. However, LSTM networks can still suffer from the exploding gradient problem.

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. 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.

CNN's are regularized versions of multilayer perceptron. Multilayer perceptron usually means fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks makes them prone to overfitting data. Typical ways of regularization, or preventing overfitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) CNN's take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using smaller and simpler patterns embossed in their filters. Therefore, on a scale of connectivity and complexity, CNNs are on the lower extremity.

A convolutional neural network consists of an input layer, hidden layers, and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final convolution. In a convolutional neural network, the hidden layers include layers that perform convolutions. Typically, this includes a layer that performs a dot product of the convolution kernel with the layer's input matrix. This product is usually the Frobenius inner product, and its activation function is commonly ReLU. As the convolution kernel slides along the input matrix for the layer, the convolution operation generates a feature map, which in turn contributes to the input of the next layer. This is followed by other layers such as pooling layers, fully connected layers, and normalization layers.

This involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction.

CNN LSTMs were developed for visual time series prediction problems and the application of generating textual descriptions from sequences of images or sequences of a text document.

We will be analyzing the performance of different architecture of the combination of CNN + LSTM networks.

### **Model-1**

It is a simple LSTM model with 100 features. We observe that its validation accuracy is 86% though we have the training accuracy 2-3% higher. There is a good amount of oscillation in the accuracy during training and there is a steep change in the curve at a few points.

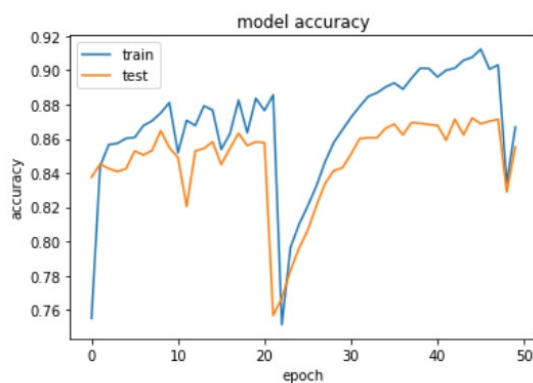
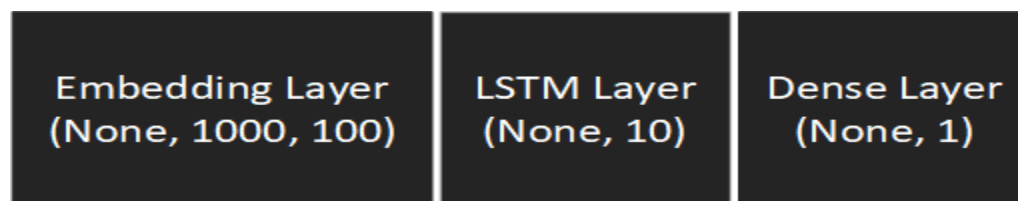


Figure 3.1

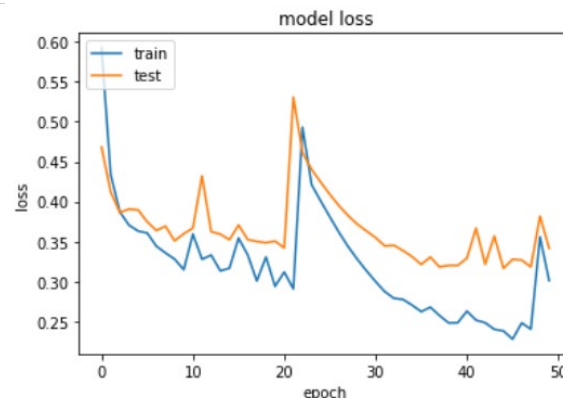


Figure 3.2

### **Model-2**

This is an upgrade of the previous model with a greater number of features and neurons. It has a significant increase in accuracy but takes a large amount of computation power due to the large network.

Embedding Layer  
(None, 1000, 300)

LSTM Layer  
(None, 50)

Dense Layer  
(None, 1)

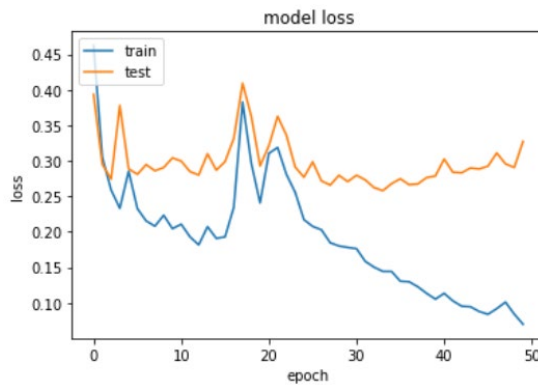


Figure 3.3

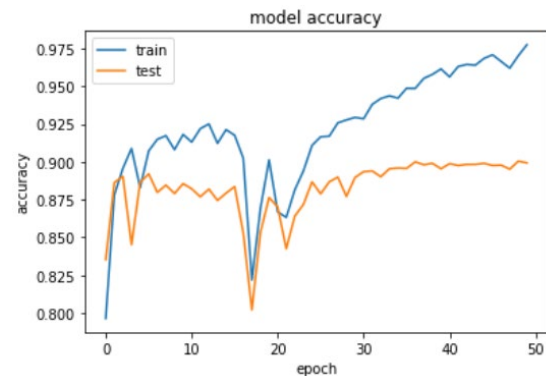


Figure 3.4

### Model-3

It is an upgrade of the previous model with an additional batch normalization layer. On adding batch normalization, it has boosted the overall accuracy of the previous model.

Embedding Layer  
(None, 1000, 300)

LSTM Layer  
(None, 50)

Batch Norm  
(None, 50)

Dense Layer  
(None, 1)

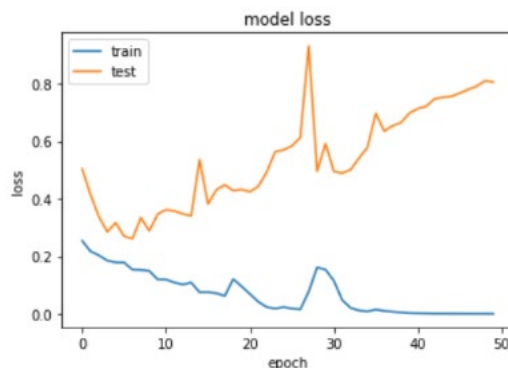


Figure 3.5

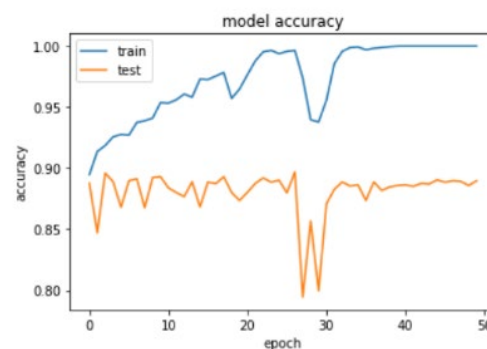
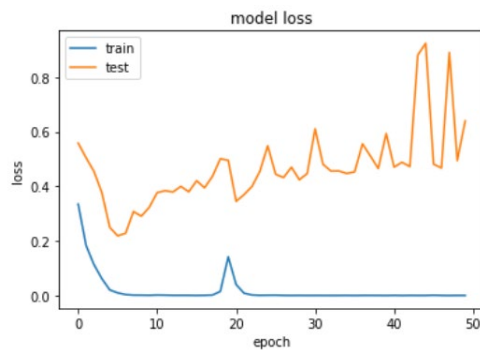


Figure 3.6

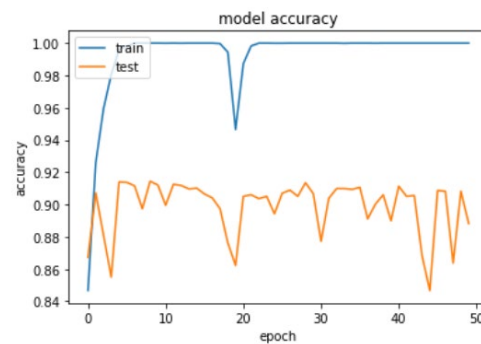
## **Model-4**

It is an upgrade of the previous model with an additional CNN network. CNN input is used for detecting the features and then more refined details are passed to LSTM boosting its accuracy to 91%. Though we have used batch normalization it is not required that will be clear from our final model.

Embedding Layer (None, 1000, 300)	Conv1D Layer (None, 1000, 25)	MaxPool Layer (None, 500, 25)	LSTM Layer (None, 50)	Batch Norm (None, 50)	Dense Layer (None, 1)
--------------------------------------	----------------------------------	----------------------------------	--------------------------	--------------------------	--------------------------



*Figure 3.7*

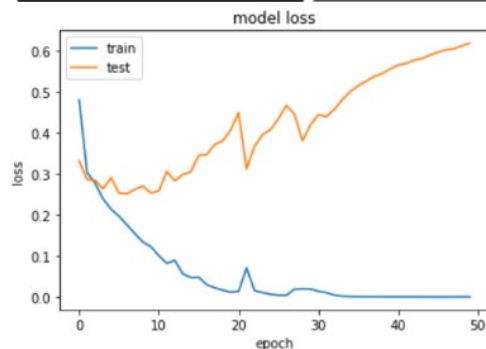


*Figure 3.8*

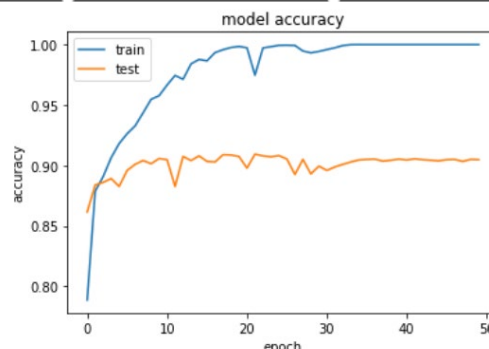
## **Model-5**

It is a downgrade of the previous model with a decrease in the number of neurons and features and removal of batch normalization. We can observe that our final model uses much less computing power and features for the same accuracy(as of only LSTM Models). Batch normalization is not required as the LSTM layer is sufficient.

Embedding Layer (None, 1000, 100)	Conv1D Layer (None, 1000, 25)	MaxPool Layer (None, 500, 25)	LSTM Layer (None, 25)	Dense Layer (None, 1)
--------------------------------------	----------------------------------	----------------------------------	--------------------------	--------------------------



*Figure 3.9*



*Figure 3.1*

### 3.3 RNN + Attention Models

Improved RNN models such as Long Short-Term Memory networks (LSTMs) enable training on long sequences overcoming problems like vanishing gradients. However, even the more advanced models have their limitations and researchers had a hard time developing high-quality models when working with long data sequences. In machine translation, for example, the RNN has to find connections between long input and output sentences composed of dozens of words. It seemed that the existing RNN architectures needed to be changed and adapted to better deal with such tasks.

Attention is a mechanism combined in the RNN allowing it to focus on certain parts of the input sequence when predicting a certain part of the output sequence, enabling easier learning and of higher quality. The combination of attention mechanisms enabled improved performance in many tasks making it an integral part of modern RNN networks.

Further, to improve the accuracy we will be analyzing the performance of different architecture of a combination of LSTM + Attention layer networks.

#### Model-6

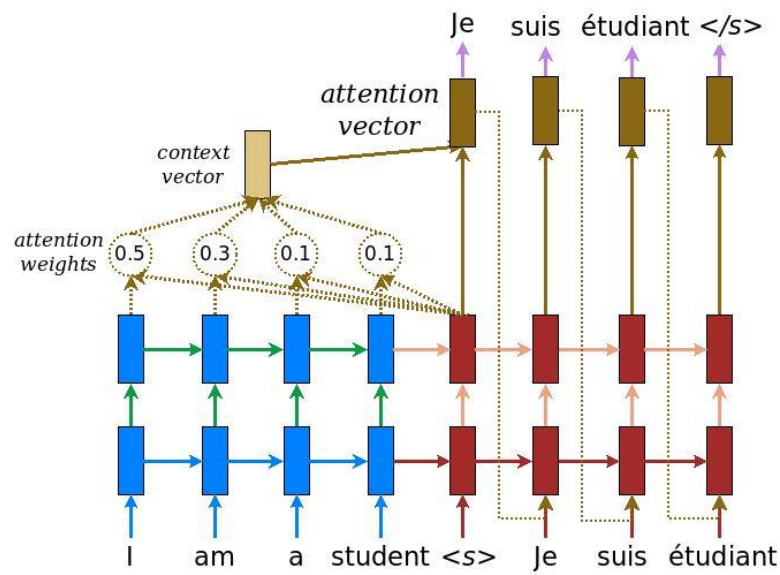


Figure 3.11

The model with an attention layer gets a significant improvement in accuracy. The attention layer decides efficiently about the data model needs to focus more on. The model is more complex as it uses a greater number of layers than previous ones, but the computational cost for the attention layer is less than recurrent layers.



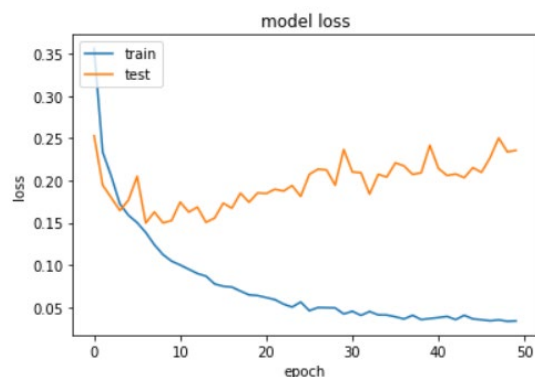
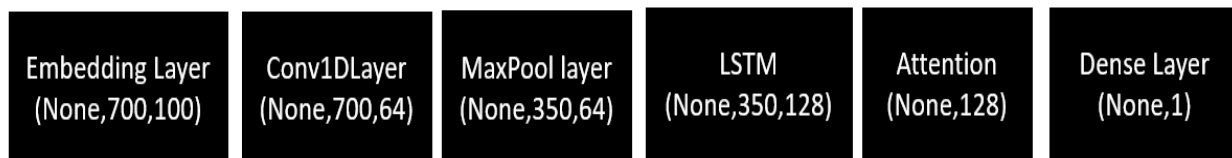


Figure 3.12

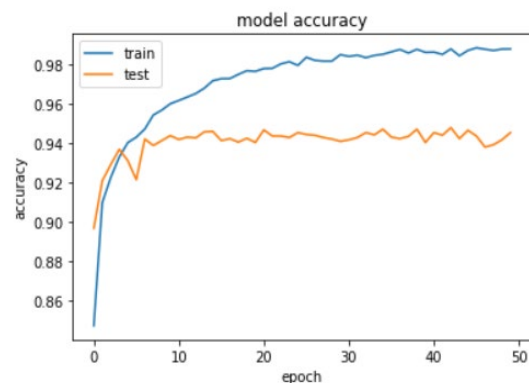


Figure 3.13

### 3.4 Transformer Model

Attention mechanisms let a model draw from the state at any preceding point along the sequence. The attention layer can access all previous states and weights them according to a learned measure of relevancy, providing relevant information about far-away tokens.

A clear example of the value of attention is in language translation, where context is essential to assign the meaning of a word in a sentence. In an English-to-French translation system, the first word of the French output most probably depends heavily on the first few words of the English input. However, in a classic LSTM model, to produce the first word of the French output, the model is given only the state vector of the last English word. Theoretically, this vector can encode information about the whole English sentence, giving the model all necessary knowledge. In practice, this information is often poorly preserved by the LSTM. An attention mechanism can be added to address this problem: the decoder is given access to the state vectors of every English input word, not just the last, and can learn attention weights that dictate how much to attend to each English input state vector.

When added to RNNs, attention mechanisms increase performance. The development of the Transformer architecture revealed that attention mechanisms were powerful in themselves and that sequential recurrent processing of data was not necessary to achieve the performance gains of RNNs with attention. Transformers use an attention mechanism without an RNN, processing all tokens at the same time and calculating attention weights between them in successive layers.

## Model-7

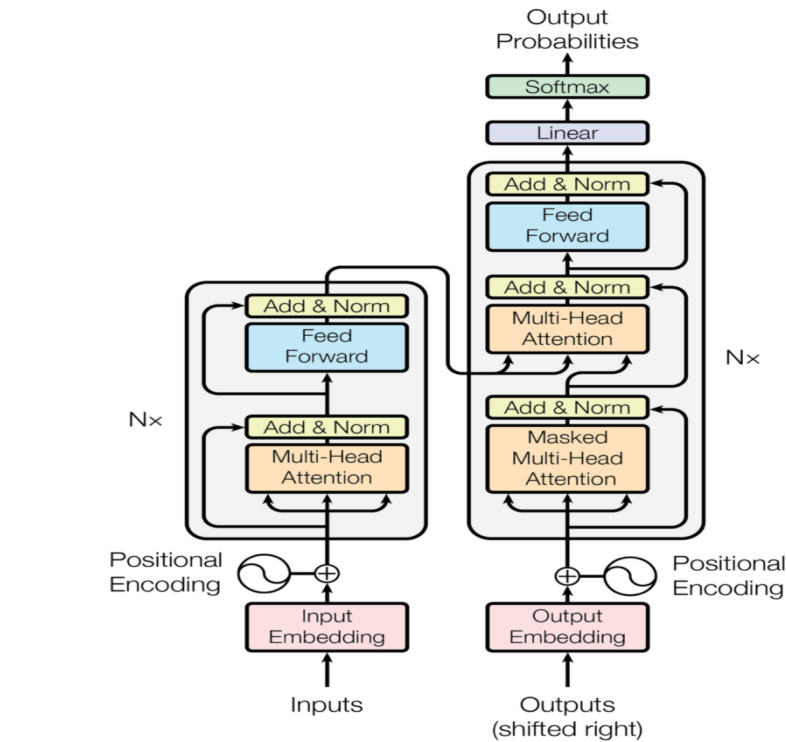


Figure 3.14

The use of XLM-Roberta has boosted the accuracy significantly. XLM models specialize in finding hidden sentiments in the texts which led to the best result. The model used is a transformer model and takes more RAM and computational cost.

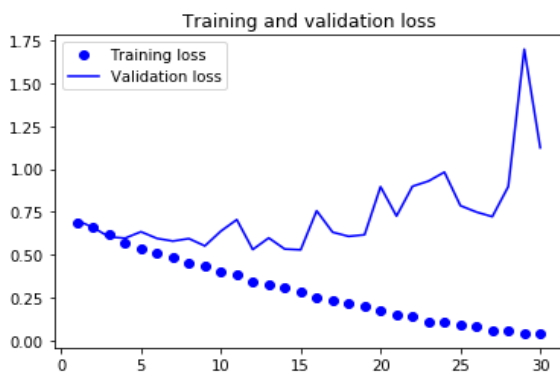


Figure 3.15

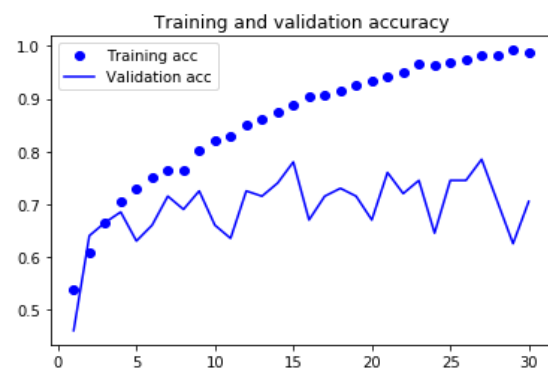


Figure 3.16

# 4

## Results and Discussion

### 4.1 Result

Model	Val. Acc.	Train Acc.	Val. Loss	Train Loss	Test Acc.
<b>LSTM only(Model-1) (100 features)</b>	86%	88%	0.30	0.35	84%
<b>LSTM only(Model-2) (300 features)</b>	89%	94%	0.12	0.26	86%
<b>LSTM only(Model-3) (300 features)(Batch Normalization)</b>	89%	98%	0.50	0.10	88%
<b>LSTM + CNN (Model-4) (300 features) (Batch Normalization)</b>	91%	99%	0.50	0.05	90%
<b>LSTM + CNN (Model-5) (100 features)</b>	90%	99%	0.45	0.05	89%
<b>LSTM + Attention (Model-6) (100 features)</b>	94%	99%	0.20	0.04	94%
<b>XLM-RoBERTa (Model-7) (100 features)</b>	98%	99%	0.16	0.02	99%

Table 4.1

### 4.2 Discussion

An important feature for predictive models that increases their applicability in multiple tasks, is their ability to generalize across datasets and tasks. Model generalization refers to the ability of a pre-trained model to handle unseen data and is mostly relevant to the model complexity and training. The majority of deep neural networks are trained on a dataset and evaluate their performance usually on a different subset of the same dataset.

The experimented models here are known to be the best for textual data and can handle long Sequential data. Recurrent neural networks are trained as a baseline model.

Long short-term memory cell helps to improve the accuracy of the model. We achieved accuracy up to 90% with the combination of RNN and CNN layers.

In the final set of experiments, the proposed Roberta transformer model is trained on 20,387 samples and tested on 5,127 headlines and articles. The training is performed using freely available TPU on Kaggle. The training takes 30 minutes to run epochs on the 'Fake News' dataset using pre-trained word embedding which is per the desired model and to show the classification results. Before moving to use the transformer model, we have experimented with various combinations of recurrent, convolution, and attention layers. By analyzing all the results, one can conclude that using the Transformer model is more effective as it significantly improved the accuracy. The presented model outperforms all other models by producing an accuracy of 99.5%. The detailed statistical results of our proposed model are shown in Table. The statistical significance ensures that one can easily classify any news as fake or legitimate using our proposed model. The train and test, accuracy, and loss are shown in the Result table,

# 5

## Conclusion and Future Work

### 5.1 Conclusion

In our study, we started with simple LSTM models and tested their performance which was proportional to the number of features and the size of the network. But still, it produced maximum accuracy of up to 89%. We observed that LSTM + CNN models are capable of producing accuracy up to 90% with a much smaller network compared to the simple LSTM model. We used the RNN + Attention model which has an accuracy of about 95%. While studying the attention model further we found that the attention model is sufficient to give the result. Then we switched to transformer models which have accuracy up to 99%. In our study, we have been able to find out how gradually we progressed from LSTM models to transformer models. Our study will help future researchers to understand how these models are derived from their predecessor models and what improved their performance from their predecessor.

### 5.2 Future Work

We are planning to work on detecting fake news shown in form of videos. We will be using our knowledge of text-based fake news detection and speech-to-text conversion. We will convert the speech in the video into text, and then try to predict whether the news is fake or not. We may develop an algorithm to identify the fake speaker and then warn the users against him/her. We have not yet created a data pipeline for our models. Our next work will contain a data pipeline to automate the entire process of fetching data and converting it to the required form. We will need to create the dataset as we have not found a suitable dataset for detecting fake news shown in form of videos.

# List of Abbreviations

<b>NN</b>	Neural Network
<b>LSTM</b>	Long Short-Term Memory
<b>GRU</b>	Gated Recurrent Unit
<b>Val.</b>	Validation
<b>Acc.</b>	Accuracy

# References

- [1] Xinyi Zhou and Reza Zafarani. 2020. A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. *ACM Comput. Surv.* 53, 5, Article 109 (September 2020). <https://doi.org/10.1145/3395046>
- [2] Ayat Abedalla, Aisha Al-Sadi, and Malak Abdullah. 2019. A Closer Look at Fake News Detection: A Deep Learning Perspective. *ICAAI Proceedings of the 2019 3<sup>rd</sup> International Conference in Artificial Intelligence* (October 2019). <https://doi.org/10.1145/3369114.3369149>
- [3] Jibrán Fawaid, Asiyah Awalina, and Rifky Yunus Krishnabayu . 2021. Indonesia's Fake News Detection using Transformer Network. *SIET. 6<sup>th</sup> International Conference on Sustainable Information Engineering and Technology* (September 2021) <https://doi.org/10.1145/3479645.3479666>
- [4] Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari. 2018. Fake News Identification on Twitter with Hybrid CNN and RNN Models. *SMSociety. 9<sup>th</sup> International Conference on social media and Society.* (July 2018). <https://doi.org/10.1145/3217804.3217917>
- [5] S. I. Manzoor, J. Singla, and Nikita, "Fake News Detection Using Machine Learning approaches A Systematic Review," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 230-234, <https://doi.org/10.1109/ICOEI.2019.8862770>.
- [6] Zhao, Zhe & Resnick, Paul & Mei, Qiaozhu. (2015). Enquiring Minds: Early Detection of Rumours in social media from Enquiry Posts. 1395-1405. <https://doi.org/10.1145/2736277.2741637>.
- [7] Sa, Ahmed & hinkelmann, knut & Corradini, Flavio. (2020). Development of Fake News Model using Machine Learning through Natural Language Processing.
- [8] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi and B. -W. On, "Fake News Stance Detection Using Deep Learning Architecture (CNN-LSTM)," in *IEEE Access*, vol. 8, pp. 156695-156706, 2020, <https://doi.org/10.1109/ACCESS.2020.3019735>.
- [9] [arXiv:1906.05659](https://arxiv.org/abs/1906.05659) [cs.CL]
- [10] M. U. Salur and I. Aydin, "A Novel Hybrid Deep Learning Model for Sentiment Classification," in *IEEE Access*, vol. 8, pp. 58080-58093, 2020, <https://doi.org/10.1109/ACCESS.2020.2982538>.