A Comparative Study of Fake News Detection Using Machine Learning and Deep Learning Techniques

MASTER OF TECHNOLOGY

in

SOFTWARE ENGINEERING

By

NEVIL DOLPHY DSOUZA

(2K22/SWE/06)

Under the supervision of

Mr. Rahul
Assistant Professor
DTU

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi – 110042. India

MAY, 2024

ACKNOWLEDGEMENT

I am grateful to Mr. Rahul (Asst. Professor, Dept. of Software Engineering) and all of the Department of Software Engineering faculty members at DTU. They all gave us a lot of help and advice for the project.

I'd also want to thank the University for providing us with the laboratories, infrastructure, testing facilities, and environment that allowed us to continue working without interruption. I'd also like to thank our lab assistants, seniors, and peer group for their aid and knowledge on a variety of subjects.

Nevil Dsouza

2K22/SWE/06

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE’S DECLARATION

I Nevil Dsouza, 2K22/SWE/06 of Master of Technology (Software Engineering) hereby certify that the work which is being presented in the thesis entitled “**A Comparative Study of Fake News Detection using Machine Learning and Deep Learning Techniques**” in partial fulfilment of the requirements for the award for the Degree of Master of Technology, submitted in the Department of Software Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from January 2024 to May 2024 under the supervision of Mr. Rahul.

The matter presented in the thesis had not been submitted by me for the award of any other degree of this or any other Institute.

NEVIL

Candidate’s Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

27/5/24

Signature of Supervisor

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR

Certified that **Nevil Dsouza** (2K22/SWE/06) has carried out their search work presented in this thesis entitled **"A Comparative Study of Fake News Detection using Machine Learning and Deep Learning Techniques"** for the award of **Master of Technology** from Department of Software Engineering, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

27/5/24

Date: 27/05/2024

Signature
Mr. Rahul
Assistant Professor
Department of Software Engineering

ABSTRACT

The term "fake news" describes information that has been purposefully falsified or misrepresented and is presented as authentic journalism to mislead or control viewers. The rampant increase in misinformation and fake news across social platforms has created a multitude of problems across various spheres of society, affecting individuals, communities, and even global affairs. News plays an important role in our lives by providing us with current information from all over the world. With the rise in popularity of online news, there has been a complete change in how we consume news media. So, with the increasing popularity of social platforms along with the ease with which misinformation can be spread, a way to detect fake news has been paramount. In this study we have studied and analysed various papers related to detection of fake news using machine learning and deep learning models. The aim of the study is to evaluate and compare the performance of various algorithms in detecting fake news. The study compares six algorithms namely, Logistic Regression, XGBoost algorithm, Naive Bayes, Recurrent Neural Networks (RNNs), Long Short-term Networks (LSTMs), and Gated Recurrent Unit Networks (GRUs). In our research, we found that LSTM had the best performance for the WELFake dataset.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

Abbreviation	Long Form
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
LR	Logistic Regression
XGBoost	eXtreme Gradient Boosting
CM	Confusion Matrix

CHAPTER 1

INTRODUCTION

1.1 Overview

News and News media play a crucial role in today's society, everyone receives new information on a topic either through traditional news media or through online news sites. News is one of the few ways we can learn about different perspectives, ideas, and cultures. It also provides us with the latest information about events, such as economic trends, political changes, etc. In recent years, due to the rise in the usage of social media, most of the users get most of their news from a social platform. With the ease of sharing information through social platforms, an influx of fake news and misinformation in our daily feeds has been unavoidable. This mix of disinformation along with actual information on social platforms has created a real problem for the people to distinguish one from another. Compared to factual news, fake news typically tends to be more sensational and emotionally charged. It is more likely to manipulate people's emotions, evoking intense responses such as rage, fear or excitement. This emotional appeal also makes it more likely to spread fake news faster by individuals without proper fact-checking.

With the rapid rate at which news comes, it can sometimes be difficult for the user to keep a track of which sources are more credible or which articles are disinformation. A challenging issue has emerged due to the proliferation of false information purporting to be trustworthy news articles, which are fundamentally connected to current communication systems. The complexity lies in the deliberate crafting of misleading narratives, fabricated stories, and distorted facts that cloak themselves within the semblance of genuine news, exploiting the vulnerabilities of audiences and digital platforms. As such, the need for advanced detection mechanisms capable of discerning the authenticity of news content has never been more pressing.

To curb the increase in widespread sharing of misinformation by individuals, they must know which articles are factually not true. With social media being at the grasp of millions of users, it is undeniably important for the platforms to have some kind of detection mechanism for fake news being shared on their platform. A platform can handle this in multiple ways, for example, they can only allow articles/links from credible source, or have a dedicated team to weed out misinformation for the platform. Both of these solutions are not good for the long term, as there will be a time when some niche news can only be found from unverified sites or the platform grows so big the it becomes infeasible for them to manually detect and remove all the misinformation. A better way to detect and manage fake news is by using an automatic fake new detection system. Fake new detection algorithms are used to find and classify news articles from social platforms by analysing the article's context. Fake news detection algorithms is similar to text classification, which is a classical problem of natural language processing. This detection method provides us with a way to automatically verify a article's credibility without the need for manual intervention.

1.2 Machine Learning and its types

In these past few years, there has been an immense growth in AI from simple prediction models to advanced generative images. Most of these models rely on machine learning as an indispensable component in their development. Unlike traditional programming, where explicit commands are used for every action, ML analyses and interprets data to identify patterns, improve from experience, and adapt autonomously to new situations. There has been an exponential growth in data and data collection in the past few decades. This growth in data has profoundly impacted machine learning, helping fuel its development and expanding its applications across various domains. Tom Mitchell defines ML [1] as: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." ML can use a variety of data types, from structured data like an excel sheet to unstructured data like an image, where each serves different purposes and has different applications. Data available to the ML algorithm plays a role in deciding which machine learning types are to be used. Generally, Machine learning can be basically classified as: 1. Supervised, 2. Unsupervised, 3. Semi-supervised, and 4. Reinforcement Learning.

1.2.1 Supervised Learning

Supervised learning is considered to be one of the most used ML techniques, where the labelled training data is used for making predictions. If we had to compare supervised learning to a real-life use case, then it would be akin to a teacher-student relationship, in which the “teacher” teaches the “student” (algorithm) with examples (training data) and its corresponding response (labels). The end goal for supervised learning is to map the inputs to the corresponding output, such that the mapping can predicting results for unseen data as accurately as possible.

1.2.2 Unsupervised Learning

Unsupervised learning is yet another popular ML technique, which discerns new patterns and basic structure from input data that doesn’t have their corresponding output available. Unlike supervised learning, there are no explicit input-output pairs provided to the algorithm, so it is much harder to train unsupervised algorithms. These algorithms are good when we don’t have clear objective and want the learning to be more exploratory

1.2.3 Semi-Supervised Learning

Semi-supervised learning incorporates elements of both unsupervised and supervised learning techniques. In this technique, the data provided has both labelled and unlabelled data to be trained on. Semi-supervised learning takes advantages from both supervised and unsupervised learning to create a better model. By using unlabelled data, the algorithm aims to improve the model’s capability to create a generalised solution. And by using labelled data, it aims to have a model with better accuracy. This approach is usually used in domains where obtaining purely labelled data can be extremely expensive or time consuming.

1.2.4 Reinforcement Learning

Unlike all the previous types of machine learning techniques, reinforcement learning (RL) doesn’t require the traditional dataset to make a model. RL is considered as subfield of ML technique in which a special entity called the agent is used to learn by making decisions on the basis of their interaction with the environment to reach a specific goal. This agent learn by taking actions based on feedback received in the form of a penalty or reward. The main object of this technique is to maximise the rewards received over the course of training. Reinforcement learning has various applications in the field of robotics, AI game play, driverless driving, etc.

1.3 Deep Learning

According to [2, 3], “deep learning is a class of machine learning algorithms that: (1) use a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input, (2) learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.”. The neural networks are built on the human brain structure and function; precisely, an artificial neural network is built using biological neuron as its basis. There are various neural network models from the basic perceptron to the more advanced Bidirectional LSTM, which uses these same neurons as their base in each layer of the network. For a neural network to be considered a deep neural network, there should be at least 3 basic layers. These layers are:

1. Input Layer

   This layer acts as an entry point for the neural network model. Each neuron in this layer represents one aspect of the input data.

2. Hidden Layer

   This layer is where most of the computation for the given model happens. Upon receiving data from the input layer, it is processed and modified by this layer before being passed to the output layer. There can be one or more hidden layer in a given model. The layer is called “hidden” because these layers are not directly visible from the input or output.

3. Output Layer

   This layer is where the final output or prediction from the network is presented at. It usually takes the transformed data from the hidden layer and maps them to their corresponding output format.

By mimicking how biological neurons work and interact, and by adopting efficient learning mechanisms, deep learning models are effective at learning in a way not too dissimilar to the way humans brain functions.

1.4 Fake News

Fake news is referred to misleading or deceptive information that can be presented in a way that seems genuine news. Nowadays with social media, fake news doesn’t just exist in traditional news sites, but can also be found on platforms like instagram, tiktok, etc. in the form of images and short form videos. In recent years there has been no shortage of fake news spreading through social media like wildfires, causing mass confusion among the populace. Fake news can be broadly classified into two types: misinformation, and disinformation.

Disinformation generally refers to fake or false information that has been deliberately generated and shared with the pure intention of deceiving the people. It is mostly used as a tool for propaganda or as a political manipulation tool. On the other hand, misinformation refers specifically to false information is being shared without the aim of misleading or deceiving anyone. This kind of fake news is generally propogated by people who may truly believe that the data they are sharing is true.

Fake news has a profound and multifaceted impact on everyone’s life, which may affect various aspects of life, from personal relationships to global political dynamics. Here are some impacts of fake news:

1. Fake news can lead to undermining trust in the media and public institutions as a whole. This happens when people cannot differentiate between truthful and

accurate information from false information. Fake news is one of the major reason for people to be so distrustful for traditional media and public institutes in the recent years.

2. Fake news can and has been used to create a greater political division, as people usually tend to believe news which align to their beliefs even if they are from unreliable sources.
3. False information regarding healthcare like vaccine safety or disease treatments, can lead to public panic and a health crisis. For example, in the USA during the peak of COVID-19 pandemic there were some false information passed on about the vaccine which lead to vaccine hesitancy and resistance from certain sections of the public.
4. Disinformation can cause violence by spreading fake information that exasperates religious, ethnic or other social tensions. For instance, social media rumours have led to mob violence in various parts of the world.

CHAPTER 2

LITERATURE REVIEW

In [4], **Govind Singh Mahara et al.** proposed 2 deep learning based models using RNN-LSTM, and Bi-LSTM for fake news detection. Both the approaches involved preprocessing the text, creating word embedding, and using dropout layer to ignore some nodes in the layer randomly. All the results are obtained using Adam optimizer and the Binary cross-entropy loss function. Adam Optimizer is a variation of optimization algorithm which is deployed in reducing the loss function when training neural networks. Binary Cross-entropy is the loss function used in this paper. The paper presents the results of these approaches by varying the dropout and padding length. RNN-LSTM achieves its best performance of 85% accuracy when the dropout was set to 0.2, whereas Bi-LSTM achieves 93% accuracy at dropout set to 0.2 for both padding length of 100 and 700.

In [5], **Ushashree P et al.** has developed a context-aware fake news detection model using algorithms like CNN, Decision Tree, and LSTM. The authors have generated their own dataset by using various keywords like covid , politics, and sports. The system involves data collection, data preparation, content-based analysis, and interpretation of the results. The dataset of four main components namely, date, title, article_text and label(real or fake). From the four algorithms used here, LSTM performed the best on their dataset while the other algorithms had an accuracy between 97-98%.

Shrutika S. Jadhav et al. [6], introduced a novel model named Deep Structured Semantic Model (DSSM) and used it along with LSTM for identifying and classification of news, on whether it is fake or factual. The model is created using the twitter dataset as train and test data. DSSM model achieves a maximum of 93% accuracy and LSTM achieves a 96% accuracy when the data is split 75-25 for training and test. When both these models are combined, the DSSM-LSTM model achieves a 99% accuracy on twitter dataset when the test-train split is set to 75-25.

In order to detect fake news and confirm whether it is true **Jamal Abdul Nasir et al.** [7] introduced a combined model with CNNs and RNNs in their work. ISOT and FA-KES datasets were used to train this hybrid model. ISOT dataset contains 45,000 entries in it, with equally distributed credible and fake news information, and FA-KES dataset consists of about 800 news entries about the Syrian war, with similar distribution of fake and credible news.For FA-KES dataset, the proposed hybrid model performed with 60% accuracy and for ISOT dataset, the model had a 99% accuracy.

Pritika Bahad et al.[8], presented a RNN Bi-LSTM model which can be used for detection of fake news. Author also compared their model with others like LSTM, CNN and RNN. The model is trained and evaluated using two datasets from kaggle[9][10]. Here the data from both the datasets(DS1 and DS2) are divided into 60:20:20 i.e. 60% for training the model, 20% for validating the said model and the rest for testing. This RNN Bi-LSTM model scored 91.08% accuracy on DS1 with validation accuracy of 89.74% and for DS2 dataset validation and testing accuracy of 98.25% and 98.75% was achieved.

The paper[11] written by **Mu-Yen Chen et al.** discusses COVID-19 related fake news and how it can be dealt with using deep learning algorithms. Author compares various DL model like LSTM, BiLSTM and GRU. This study explore detection of fake news by focusing both Chinese and English languages news. For English news, Fakecovid dataset[12] has been used, which consists of about 7,500 articles from Snopes and Poynter. Apart from this the paper also uses GitHub COVID-19 News-corpus dataset[13], which are collected from news channels like Fox News, BBC and CNN with a total of 5000 items. For Chinese language news, the authors used a collection of news of Covid-19 from Taiwanese media[14]. The study shows that Bi-LSTM has the highest accuracy of 99.47% with a dropout of 0.5. Highest accuracy for LSTM and GRU are 99.25% and 99.35% respectively.

In [15], author Ashwini et al. implements a very simple naïve bayes classifier to detect faux news. The authors in their research have created an android app to interact with their fake news detection model. The paper presents a simple model which uses naïve bayes classifier as their classifier model and for feature extraction they have used TF-IDF vectoriser. The classifier in this paper has an accuracy of 80%.

Authors S. S. Reddy et al. in their paper [16] have presented a model for fake news detection which uses eXtreme Gradient Boosting algorithm. This model was created using LAIR dataset which contains about 12800 data entries from website called politifact. The model performed a 98% testing accuracy on the LIAR Dataset with 70:30 data split.

CHAPTER 3

METHODOLOGY

This section of the report gives an outline of the approach used to compare and analyse detection of fake news using various algorithm. The approach consists of dataset collection, data preprocessing techniques, feature extraction process, construction of ML and DL models, and evaluation of these models. The research mainly focuses on comparing the performance of advanced DL algorithms and traditional ML models in identifying fake news.

Considering how adversely fake news affect opinions among the public, it is extremely important to create and analyse methods to decrease the effect of false information on people's opinions. This study uses and compares both ML and DL approaches to identify the strengths and weaknesses of each method.

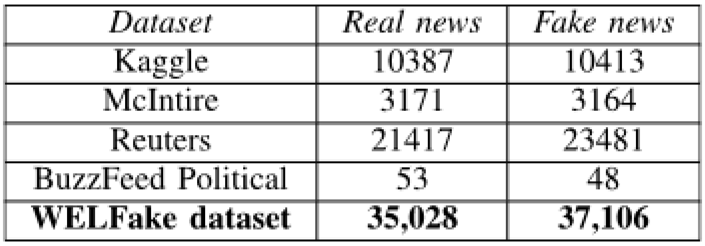
3.1 Dataset Collection

Two publicly accessible datasets, the WELFake and the dataset, were used for this study.

3.1.1 WELFake Dataset

WELFake dataset, also known as "Word Embedding Over Linguistic Features for Fake News Detection", is a dataset about fake news and presented by Pawan Kumar Verma et al. [17]. It is a repository for 72,134 articles, which were collected from various sources like Reuters, Buzzfeed, McIntire and Kaggle. The dataset is almost equally distributed into real and fake news i.e. 35,028 authentic arcticles and 37,106 fake articles can be found.

Each entry in this dataset contains 4 attributes namely, 1. Title, 2. Arctile_Text, 3. Serial No., and 4. Label(Output). For real news articles, the value of the label is set to 1 otherwise it is set to zero.



Dataset	*Real news*	*Fake news*
Kaggle	10387	10413
McIntire	3171	3164
Reuters	21417	23481
BuzzFeed Political	53	48
WELFake dataset	**35,028**	**37,106**

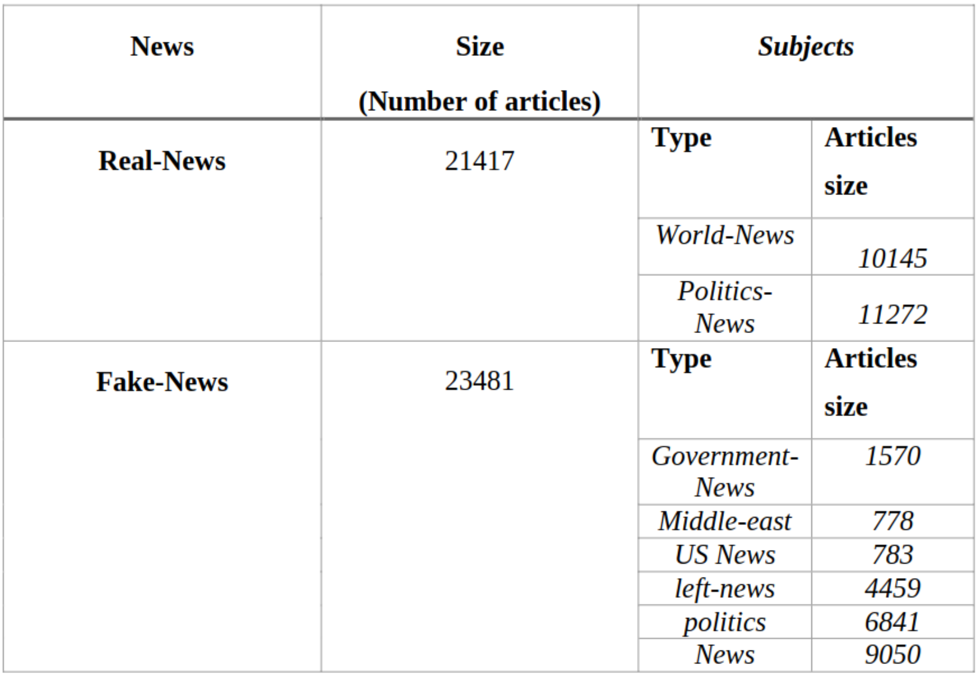
Table I: Sources of WELFake with fake/real news count

As we can observe from table I, most of the data is collected from Reuters and Kaggle.

3.1.2 ISOT Dataset

‘ISOT’ in ISOT dataset stands for “Information Security and Object Technology”. The dataset contains more than 44,000 new articles entries, collected from 2016 to 2017. The data is a culmination of all the news that are flagged by Politifact.com as unreliable and are published by legitimate news sites.

Each entry in the dataset has 4 key attributes: date, title, article_test, and label. ISOT lab has already cleaned and processed all the data, without removing mistakes that were already present in the original article. The breakdown of the ISOT dataset can be given as follows:



News	**Size** **(Number of articles)**	***Subjects***	
Real-News	21417	**Type**	**Articles size**
		World-News	*10145*
		Politics-News	*11272*
Fake-News	23481	**Type**	**Articles size**
		Government-News	*1570*
		Middle-east	*778*
		US News	*783*
		left-news	*4459*
		politics	*6841*
		News	*9050*

Table II: Data Distribution in ISOT dataset

3.2 Data Preprocessing

In this research, in order to achieve high accuracy we have performed various preprocessing steps. These steps include: 1. Removal of punctuation, 2. case conversion to lower case, 3. Tokenisation, 4. Stopword removal, 5. stemming or lemmatization.

3.2.1 Punctuation Removal

In this step, the input data is stripped out of all the unnecessary punctuation marks, which might cause problems with the accuracy of the models. We have used a pre-initialized string from the python's string library to check whether the characters in the input string is a punctuation or not.

3.2.2 Covert to lowercase

In order to abstain from countless duplication of the phrase, we transform each word in the initial text into small letters first. Usually, the case of the word has its own meaning to it which can change the result, but in the case of our research we found that multiple duplicates are more of a hindrance to performance. We used default lower function in python for case conversion

3.2.3 Tokenisation

In this step, we break down a given sentence as a group of words, such as bigram or ngram. Tokenisation is useful to remove punctuation and stopwords. This group of words is created using word_tokeniser from a predefined python library named “nltk”. Advantage of the function over a simple string split function is that it separates words from punctuation, which in turn helps in punctuation removal.

3.2.4 Removal of stopwords

In any given input string, there are many words that are common in almost all the sentence, hence providing no value in creating a fake news model. Stop-words like these needs to be removed from the sample space before training the model. NLTK library in python provides a list of stop-words which can be used to weed out unwanted words from the input space.

3.2.5 Stemming

Stemming is a way of changing a word back into its root form, generally by removing suffixes, if any. This results in words which may or may not be a valid word in the given lexicon. Aim of stemming is to decrease the total base words in a document, so that the algorithm will more efficient create a accurate model. Eg.

Consider a sentence containing words like jump, jumping and jumper, so stemming would reduce all these words into “jump”.

3.2.6 Lemmatization

Since, stemming may create an invalid word, which might not correspond to any real word in a dictionary, it is important to have words in their canonical form. Lemmatization is a process similar to stemming, in which the word is broken down into its base form, but in this case the base form is its canonical form. This means that lemmatization takes into account that the root word is a valid word in the given language. Since it converts words into valid base form, it is more resource intensive than stemming.

3.2.7 TF IDF

TF-IDF is a statistical method to convert textual data into numeric data by evaluating the importance of a given word relative to a collection of documents. It is a widely used technique for transforming data in text format into inputs that is suitable for machine learning algorithms.

TF-IDF is a combination of both TF and IDF. Term frequency can be defined as the ratio of the total count of phrase appearing in a document to total word count in the document. It is used to calculate how often a word appears in a given string or document. It is used to calculate how frequently a phrase pops up in a given string.

$$TF(t,d) \; = \frac{Number\ of\ times\ term\ t\ appears\ in\ document\ d}{Total\ word\ count\ of\ document\ 'd}$$

IDF defined as the weightage or value of a given term by checking how frequently it appears in a collection of documents. This means if a word or term have high occurrence in multiple documents then it has low importance. Words such as “as”, “then”, etc are present in every documents, but they do not have significant importance.

$$IDF(t) \; = log\left(\frac{Total\ number\ of\ documents}{Total\ documents\ with\ word\ 'w'\ in\ it}\right)$$

So TF-IDF can be calculated by simply multiplying TF and IDF.

$$TFIDF(t,d) = TF(t,d) * IDF(t)$$

3.2.8 Data Splitting

In this research, we divided the dataset into two parts: 1. Training, where the data is used to fit a new model and testing data, where the data is used to test the new model created. We have divided the data into 80:20 for training and test respectively. For splitting the dataset, we have utilized the train_test_split function, which is available in sklearn.model_selection library in python. We used stratified sampling technique to split the data into these two parts. This was done to ensure that both classes i.e. fake and true news was proportionately distributed among the training and testing data.

3.3 Machine Learning Model

We have analysed and compared three machine learning algorithms, namely

1. Logistic Regression
2. XGBoost Algorithm
3. SVM

3.3.1 Logistic Regression

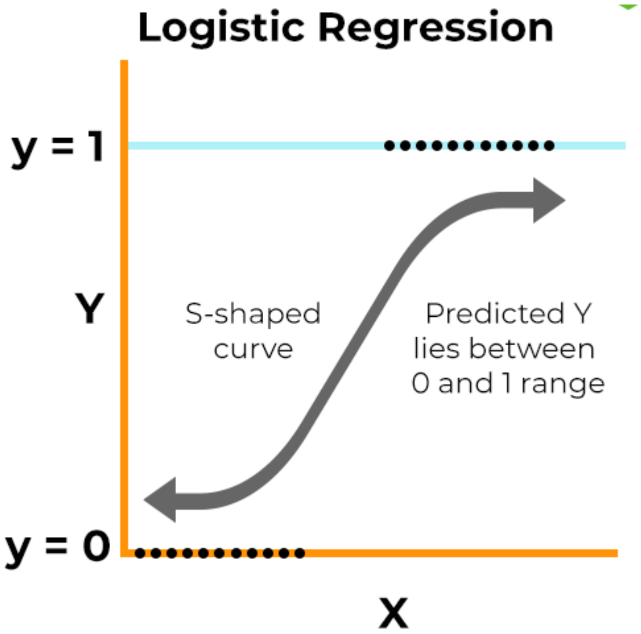


Fig 1: Logistic Regression Classifier[18]

Logistic Regression can be defined as a statistical method which is usually used for classification problems with binary outputs. In these types of classification problems, a dependent variable can be one of the two possible options, either “0” or “1”. Even though this algorithm is named “Logistic **Regression**”, it is actually algorithm for classification problems. It uses a simple sigmoid function to classify the category of a given input. Logistic Regression can be mathematically represented as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

This algorithm has been a go-to algorithm for many real life application due to its simplicity and effectiveness, though LR may not be the best suited technique for more complex,non-linear problems without any proper data transformation.

3.3.2 XGBoost Algorithm

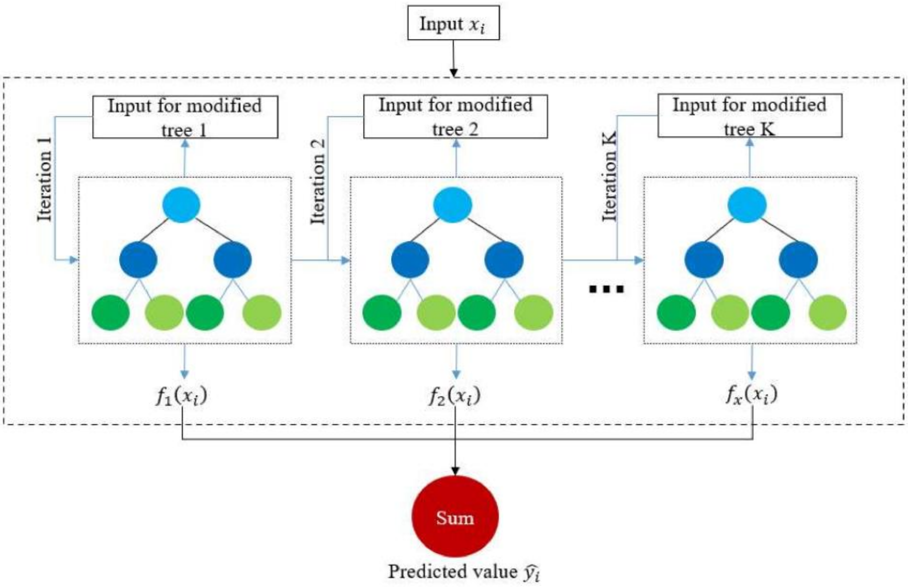

Input x_i
Input for modified tree 1
Input for modified tree 2
Input for modified tree K
Iteration 1
Iteration 2
Iteration K
$f_1(x_i)$
$f_2(x_i)$
$f_x(x_i)$
Sum
Predicted value $\hat{y}_i$

Fig 2: XGBoost Classifier[19]

XGBoost, also known as Extreme Gradient Boosting, is a type of ML based algorithm which uses optimised distributed gradient boosting framework is used to make a highly efficient, and flexible model. XGBoost basically uses multiple decision tree as its base learner and uses the one with the best performance. Thus, it can be considered a example of ensemble learning, where the prediction made by the model by combining output from multiple simpler models. XGBoost includes both the regularisation i.e. Lasso and Ridge regularisation. This regularisation helps in avoiding overfitting, by penalising more complex models. There are various advantages of XGBoost algorithm, namely,

1. Efficiency: It is designed to optimise speed and performance.
2. Scalability: XGBoost can handle dataset of large-scale with millions of instances in them.
3. Flexibility: Since XGBoost can support multiple objective function and evaluation metrics.

3.3.3 Naive Bayes

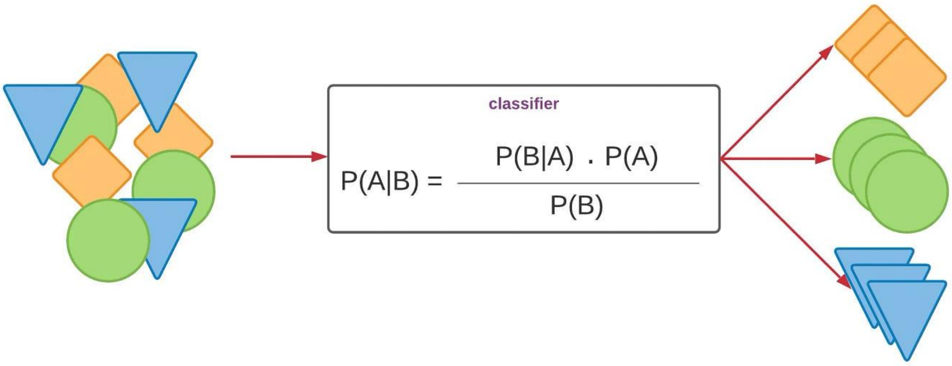


Fig 3: Naïve Bayes Classifier [20]

Naive Bayes is a type of probabilistic ML algorithm that uses Bayes' theorem as it's foundation. Even though the algorithm itself is simplistic, it works really well for categorization problems, especially when the dataset has high dimensionality.

As this algorithm is based on Bayers' theorem, the mathematical formula for naive bayes algorithm can be given by,

$$P(A\,|B) \;= \frac{P(B \mid A)\,,P(A)}{P(B)}$$

3.4 Deep Learning Models

In this research, we have performed news category classification using 3 deep learning models, namely.

1. RNN
2. Bi-LSTM
3. GRU

3.4.1 RNN

RNN can be defined as a deep learning model which incorporates some concepts of supervised learning[21]. It is extensively used in many applications involving sequential or temporal data, such as speech recognition, and NLP. RNN like many other deep learning techniques, are relatively old algorithms.

Due to their internal storage of data, RNNs can anticipate important things regarding the inputs received, which helps them to predict very precisely. RNN are able to construct a much deeper understand of a given sequence than other existing ML technique. The recurrent in RNNs is to indicate for each element of a sequence they perform the same task.

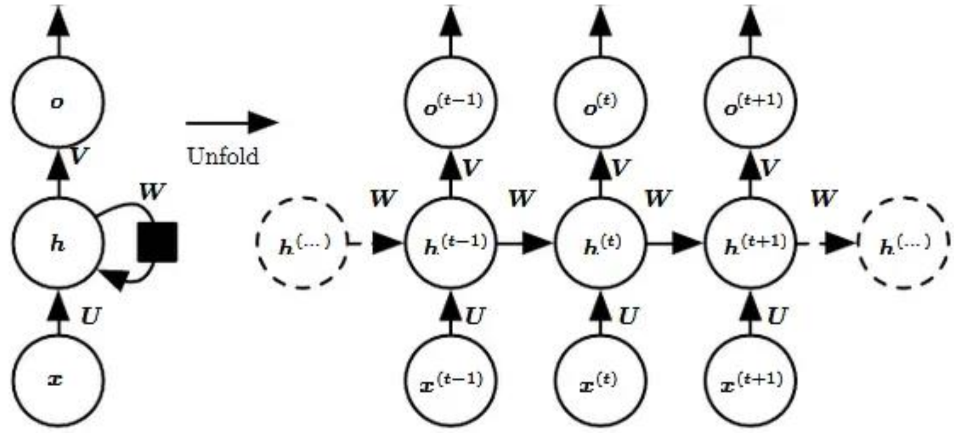


Fig 1: Unfolding of a Recurrent Layer[22]

The above figure shows how a RNN will be after its been unrolled into a complete network. Here, by unrolling we mean that we will have a network for the full sequence. For example, consider a sentence(sequence) of 15 words, then we will have a 15-layer neural network unrolled from the RNN, a single layer for each word. General step for RNN are as follows:

1. Initialisation

   To start with the implementation of RNN, we first need to initialise various parameters like, weights and biases. They can be assigned randomly or we can use pre-trained embedding.

2. Forward Pass

   This is a pretty straightforward step, her we simply calculate hidden state and output values to get the next characters using probability. This calculation is done by using the softmax equation.

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^{N} e_k^a}$$

3. Compute Loss

   Loss between actual target and predicted output can be computed using a loss function like cross-entropy or mean squared error.

$$CE = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

4. Backward Pass and update weights

   After computing loss, we will use backpropagation to modify the network variables by creating a gradient with the help of loss computed and manipulate the parameters to minimise error.
5. Iteration

   Repeat steps 2 to 5 for each element in the sequence or terminate the network once all the sequences are done.
6. Evaluation

   Performance of the implemented RNN can be measured using the test or validation dataset

3.4.2 LSTM

Long Short-Term Memory, or LSTM, is generally used for NLP based problems and can be considered as an variant of RNN with multiple different gates present. Data sequences which have long-term dependencies can be effectively dealt with by LSTMs.

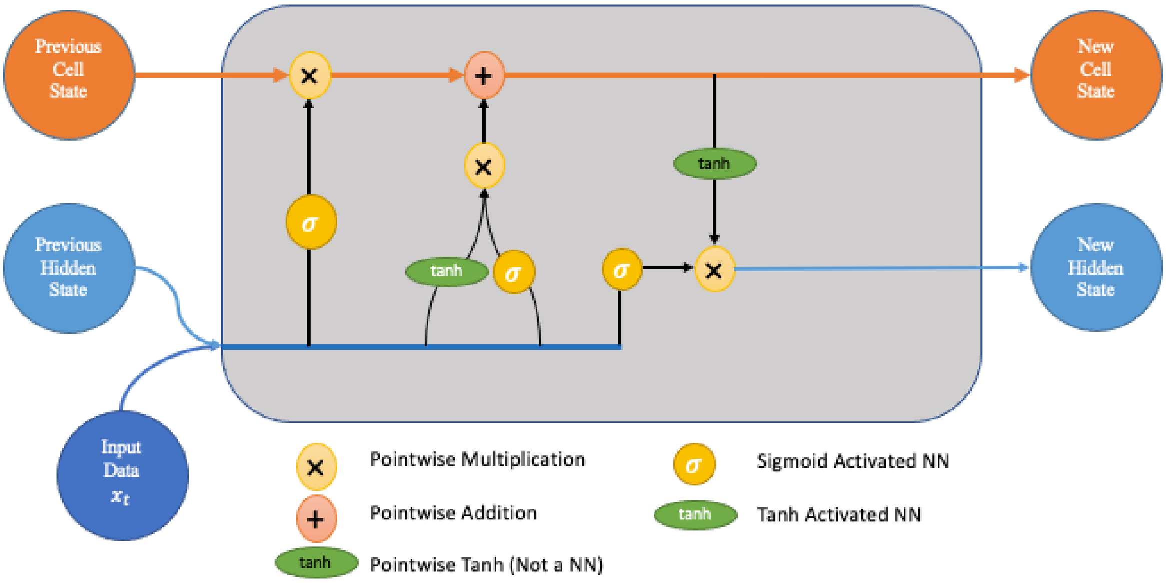


Fig 2: LSTM Unit[23]

Unlike traditional feedforward networks, LSTMs have feedback connections. LSTM networks generally consist of a chain of repeating modules, where each of these modules contains three gates and a cell state. Input, forget, and output are the three gates present in LSTM. Forget gate is used to know how much of the previous layer's output will be forgotten and how much is used in the next layer. On the basis of the current input and previous output, the input gate makes a decision of the percentage of new info to add. Remaining layer i.e. the output layer is to determine how much output should be outputted to the cell state. Sigmoid function is used as implementation for input and forget gates. The values of sigmoid function can range between zero and one, which also shows us how much info is let through. Output gate is passed through a tanh function to get value between -1 and 1.

The main advantage of LSTMs is that it can retain knowledge over extended periods of time, which is crucial for NLP jobs involving deciphering text. Problems like language modelling have complex sequential patterns and relations which LSTMs can detect and learn. But, there is a problem of vanishing gradient that can occur during training of LSTM networks. In this the gradient shrinks so small during backpropagation that it leads to ineffective learning. This problem can be solved by various techniques like gradient clipping the sets a limit to the size of the gradient.

Also, dropout regularisation can be used to avoid this problem as it randomly drops out units during training in order to avoid overfitting.

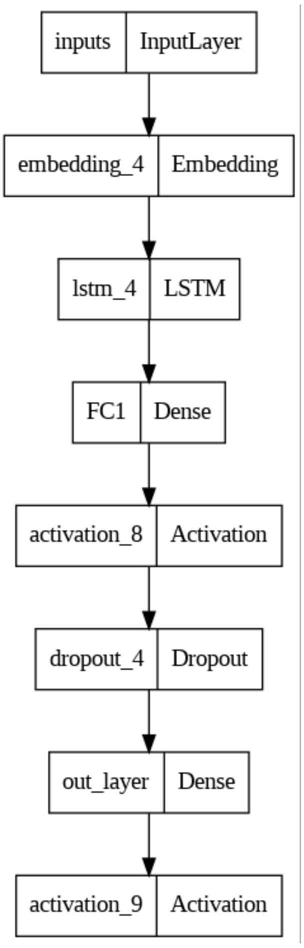


Fig 3: LSTM Model

In our implementation, initially we have performed various data preprocessing steps like feature selection, sampling data for balance dataset. After that we train our LSTM model with the training set. Our model contains various layers including the input layer, output layer, LSTM layer, Dense layer, and Dropout layer. LSTM layer is an RNN layer which can detect and learn long-term dependencies between sequence of data.. Dense layer is mainly used to learn non-linear relation between input and output. Dropout layer are used for prevention of problems like overfitting and vanishing gradient.

3.4.3 Bi-LSTM

Bidirectional LSTM is considered a extension of LSTM, in which the sequential data is processed in both directions. Bi-LSTM utilised data from both past and future context to improve the ability to identify bidirectional relationships inside of the sequences. Basically, Bi-LSTM is a RNN model which uses two LSTMs i.e. one LSTM takes input backwards and other takes it forward.

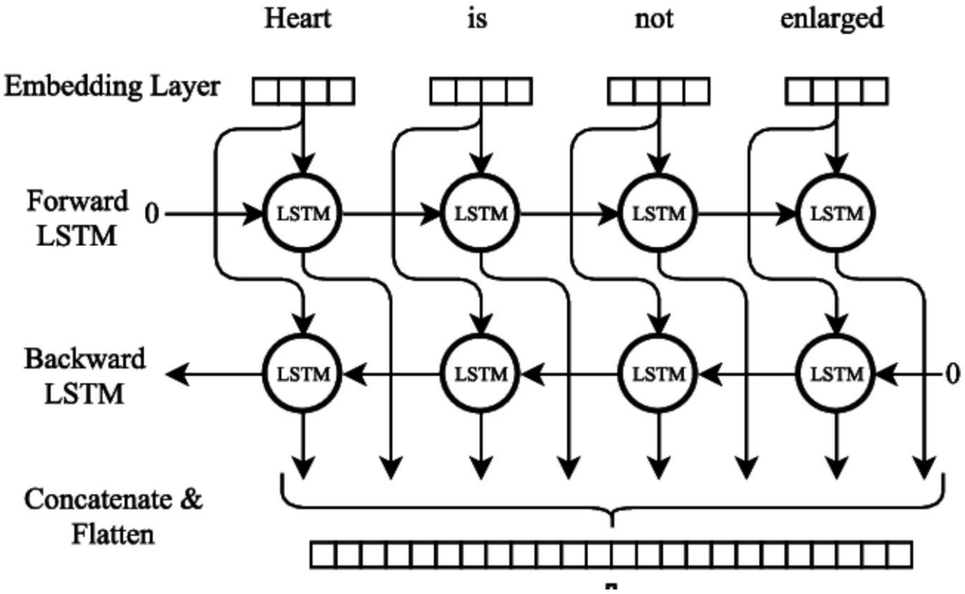


Fig 4: Bidirectional LSTM Model [24]

Steps in a Bi-LSTM:

1. Initialization
   - ○ To start implementing a Bi-LSTM, we first must initialise the network parameters i.e. their biases and weights for both the backward and forward LSTM layers. Similar to LSTM or RNN, these parameters can be randomly initialised or initialised using a pre-trained embedding.
   - ○ Set the initial hidden states and cell states to zeros or as learned parameters.
2. Forward LSTM Processing

- Starting with the forward LSTM layer, it gets the values from the input layer, which will process sequence start to end, capturing forward dependencies.

3. Backward LSTM Processing
   - At the same time as forward LSTM receiving input, the backward LSTM layer also receives the input sequence, which is then processed by the layer as a sequence in reverse, from end to start, storing the backward dependencies.
4. Merge Step
   - Merge the outputs received from both the backward and forward LSTM layers at each step, which will create a unified representation that will be a culmination of information from both directions
   - Summing, averaging or concatenation are the common merging methods to combine forward and backward hidden states.
5. Output Generation
   - If the Bi-LSTM is part of a larger network with an output layer, the merged representations can be used as input to subsequent layers for generating predictions or performing further tasks.
6. Evaluation:
   - Evaluate the performance of the trained Bi-LSTM on validation or test data to assess its ability to generalise to new, unseen sequences.

3.4.4 GRU

Gated Recurrent Unit are modified RNN model which are specially designed to overcome several drawbacks of the more conventional RNNs, and that includes difficulties in identifying and capturing long-term dependencies in sequential data. Even though GRUs provide similar capabilities as LSTMs in capturing patterns and dependencies, it is a much simpler model. It was presented by Cho et al. in 2014 as a far more straightforward substitute for the LSTM model. The main distinction between models like LSTM and GRU is in the handling of the memory cell state.

Memory cell state is managed separately from the hidden state in LSTM and its value is determined on the basis of these three gates: the output gate, input gate , and forget gate. In GRU however, “candidate activation vector” is used to replace teh memory cell state, which in turn uses two gates to update the value, namely: the update gate and the reset gate.

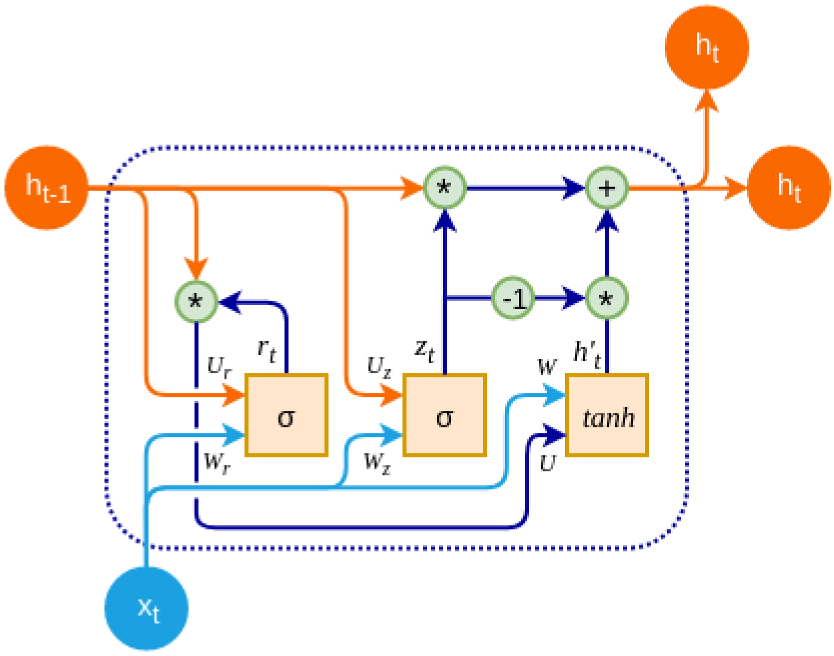


Fig 5: GRU Model[25]

Components of GRU

1. Hidden State

   The memory or data that the GRU keeps at a specific time step is represented by the hidden state.

2. Update Gate

   The amount of the new candidate activation to utilize for updating the current hidden state and the amount of the prior hidden state to keep are controlled by the update gate. The update gate uses the sigmoid activation function to make a decision regarding the relevance of the previous and new data.

3. Reset Gate

When calculating the candidate activation, the value of prior hidden gates to be ignored by the current is regulated by the reset gate.

4. Candidate Activation

   It is used to indicate any new data which can potentially be used in the current hidden state.

5. Activation Function

   Functions like sigmoid and others are utilised in GRU to regulate the gating mechanisms and scale the output of specific operations.

Steps in a Gated Recurrent Unit (GRU):

1. Initialization:

   In order to implement a GRU model, we must first initialise network parameters like, biases and weights. Pre-trained embedding or random assignment can be used for initialisation

   Set the initial hidden state to zeros or as a learned parameter.

2. Sequence Processing:

   Sequential data like words in a sentence are iterated over one element at a time.

   Input Encoding: Encode data at current timestep into a numerical representation suitable for the GRU.

3. Gate Operations:

   GRUs have 2 gates: an update gate and a reset gate.

   Update Gate Operation:

   In this gate, the amount by which the prior hidden state versus the new candidate activation should be retained needs to be ascertained by us.

   Reset Gate Operation:

   In reset gate, we compute percent of previous data is to be deleted.

4. Candidate Activation (h~th~t) Calculation:

   In this step, we calculate the candidate activation using the two gates, update gate and reset gate.

5. Hidden State Update:

   Combination of previous hidden state and the current activation is used to create the new hidden state.
6. Output Generation (Optional):

   If the GRU is part of a larger network with an output layer, the final hidden state (htht) or processed information can be used for further tasks like classification or prediction.
7. Repeat or Terminate:

   Repeat the sequence processing steps for each element in the sequence (if needed) or terminate the process when the entire sequence has been processed.

3.5 Evaluation Metrics

This part of the report focuses on how the classification model mentioned above are evaluated. Performance evaluation of models is a crucial step in analysing and comparing different models with each other. In our research we have use the follow evaluation metrics.

3.5.1 Confusion Matrix

For classification models, confusion matrix is generally used for evaluating performance by visualising data in table format. Confusion matrix is a table that showcases prediction which made by the given algorithm in comparison to ground truth. It is useful for understanding the types of errors a model can make.

3.5.2 Accuracy

Accuracy is the most frequently used performance metrics in machine learning. It is the ratio of predictions that are correctly selected to total instances. Formula for accuracy is:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

3.5.3 Precision

Accuracy is generally used to gauge the performance of a model, however, it is preferable to employ precision when the price of false positives is significant. Precision is defined as true positive divided by total positive cases.

$$Precision = \frac{TP}{TP + FP}$$

3.5.4 Recall

Sensitivity, is a metric that shows us how well a model detects a positive instance in all positive samples. Usually recall values are more useful when a false negative has a very big impact on the result.

$$Recall = \frac{TP}{TP + FN}$$

3.5.5 F1 Score

F1 score can be defined as an harmonic mean of recall and precision. This means that f1 score can be used when we need to take both false positives and negatives into consideration.

$$F1\,score = 2 * \frac{(recall * precision)}{(recall + precision)}$$

CHAPTER 4

RESULTS

The study carried out an extensive comparative analysis of the ML and DL methods used in categorization of fake news. A variety of DL models were used here, such as the GRU, LSTM, and basic RNN along with some ML algorithms like Logistic Regression, XGBoost Algorithm and Naïve Bayes. For this research we have used WELFake dataset for comparison of all the above mentioned models. For training these models, we have used the first 100 words of the article from the dataset. Following are the accuracy, precision, recall, f1-score and a confusion matrix for each model.

4.1 Confusion Matrix

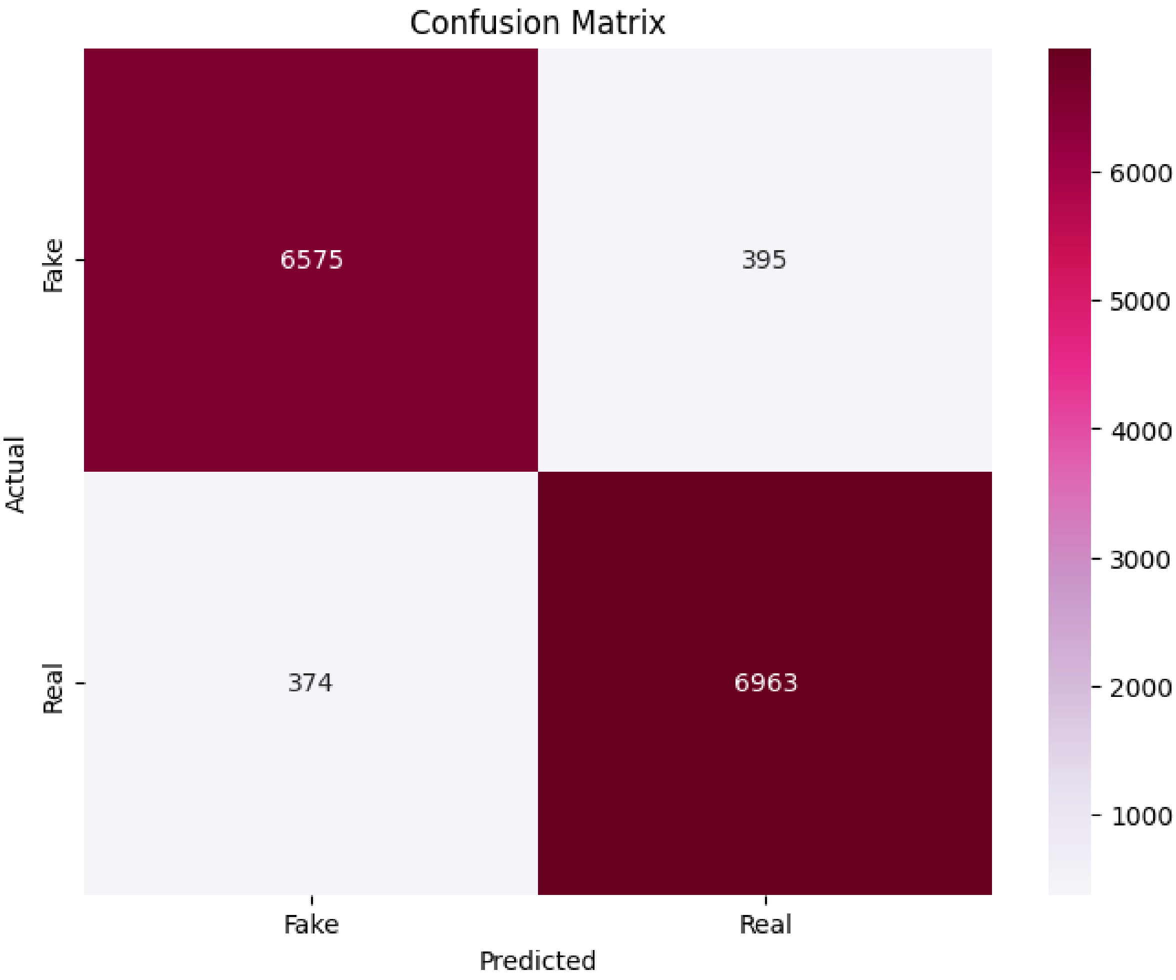


Fig: CM of Logistic Regression

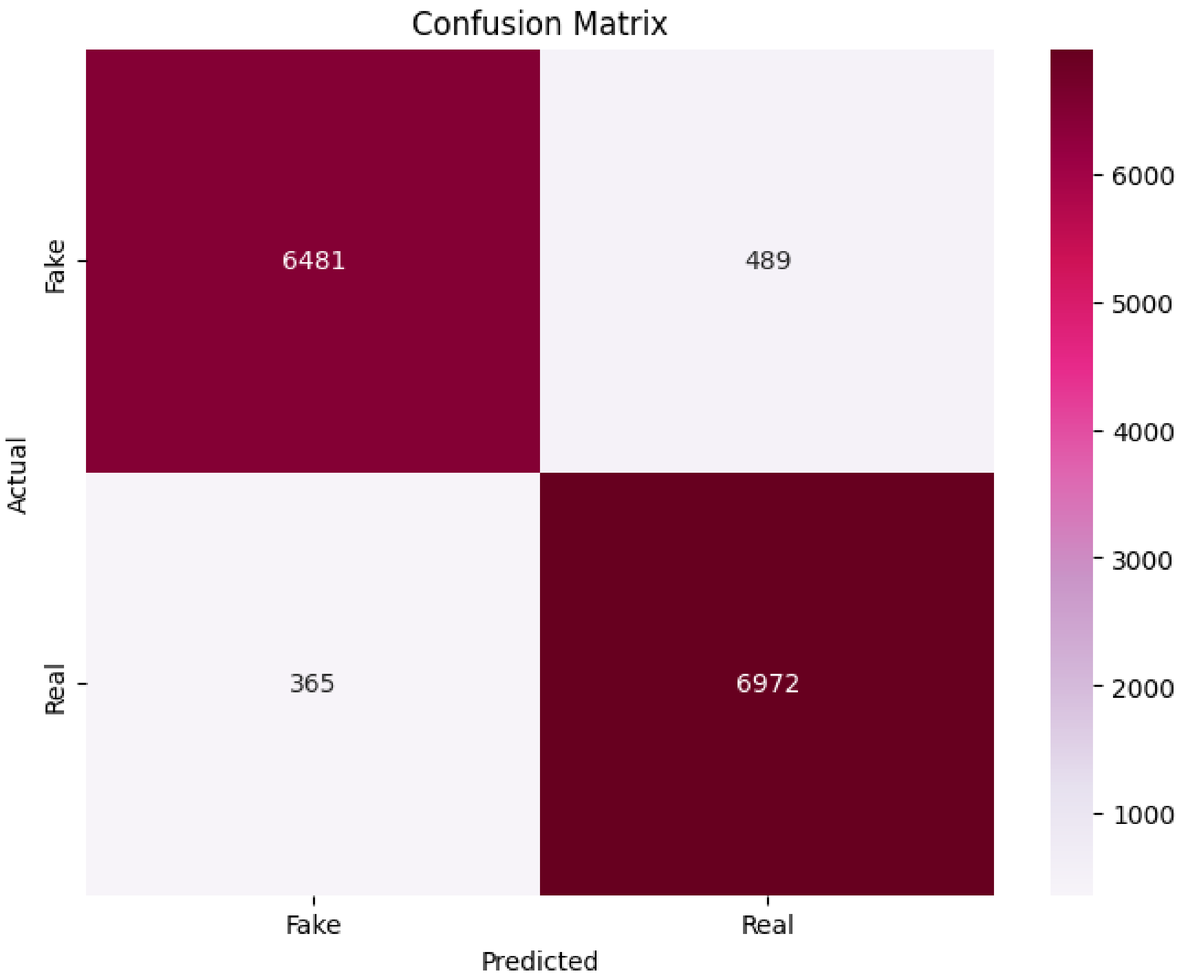


Fig: CM of XGBoost

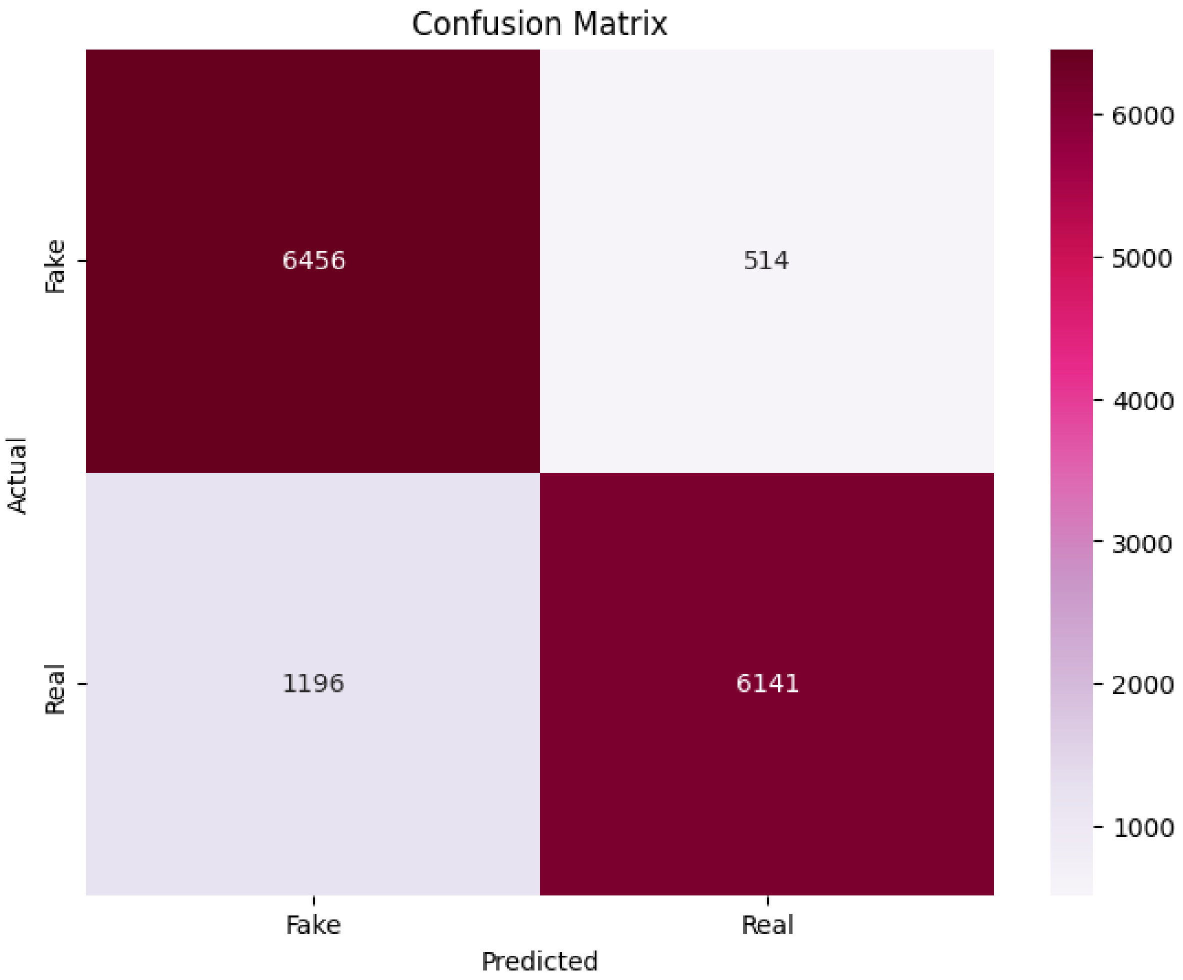


Fig: CM of Naïve Bayes

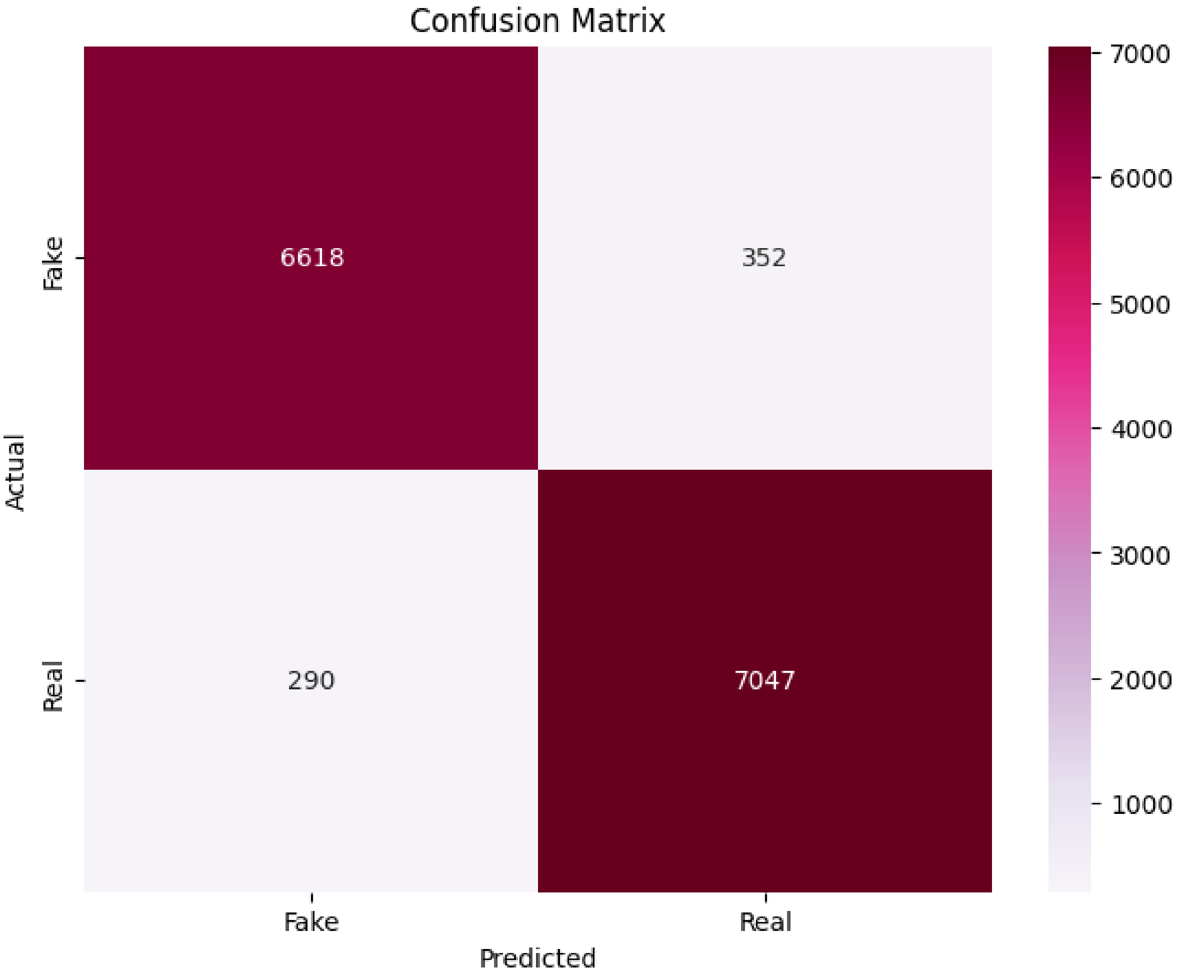

Confusion Matrix
6618
352
290
7047
Fake
Real
Actual
Fake
Real
Predicted
7000
6000
5000
4000
3000
2000
1000

Fig: CM of RNN

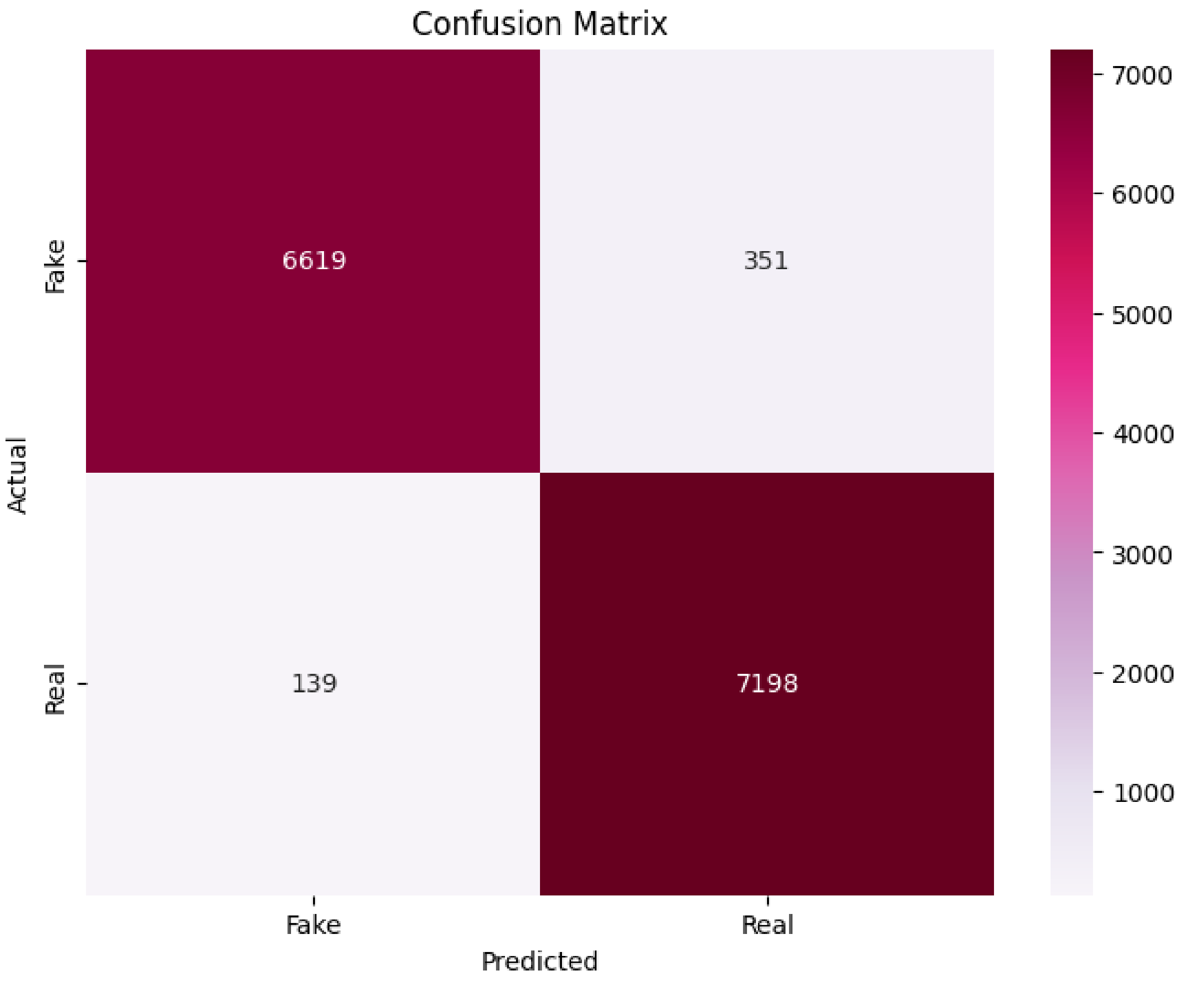


Fig: CM of LSTM

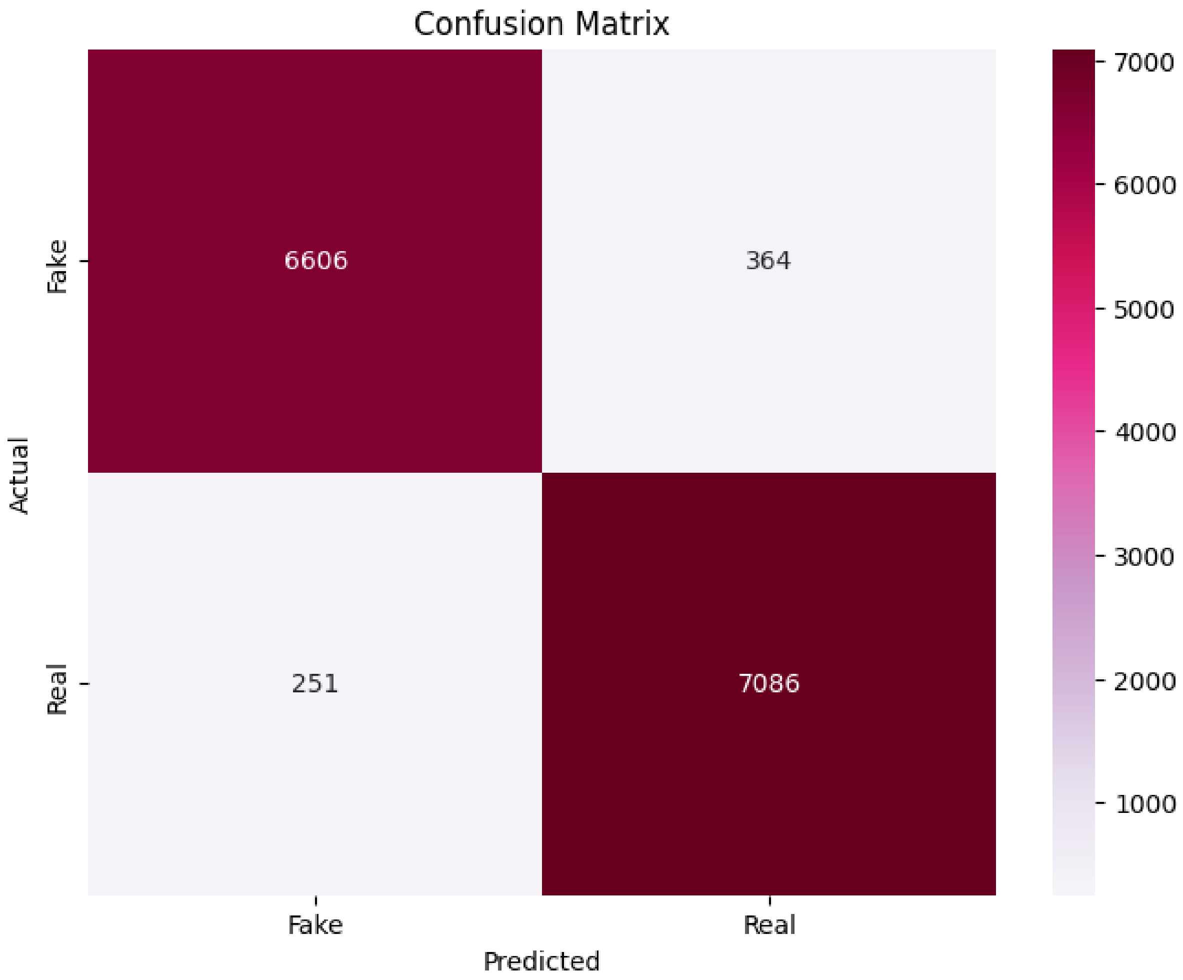
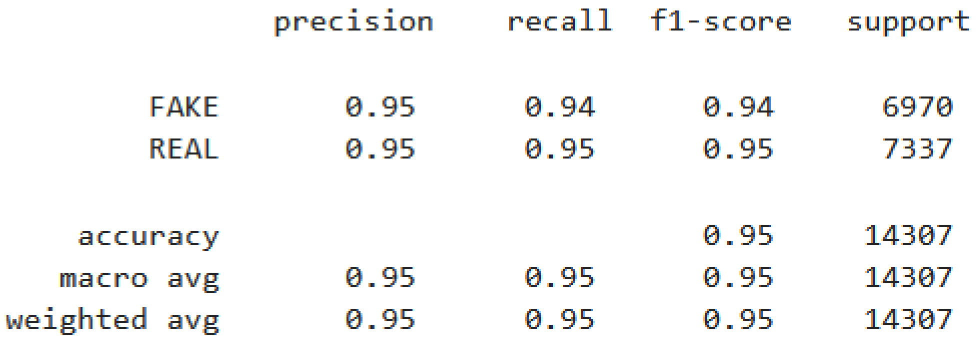


Fig: CM of GRU

4.2 Classification Report

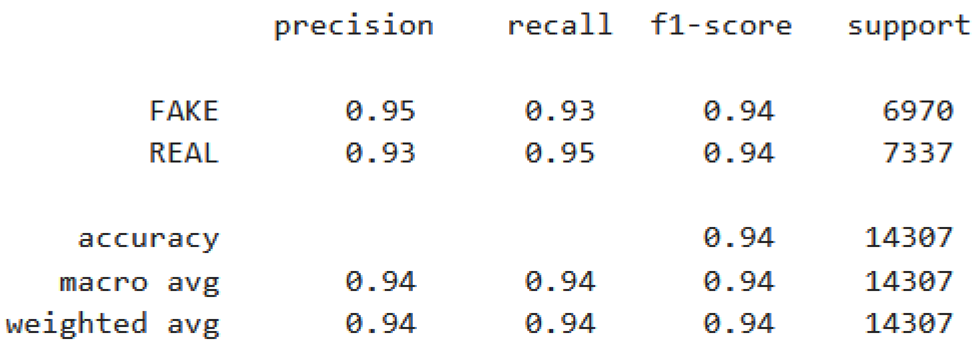


```
              precision    recall  f1-score   support

        FAKE       0.95      0.94      0.94      6970
        REAL       0.95      0.95      0.95      7337

    accuracy                           0.95     14307
   macro avg       0.95      0.95      0.95     14307
weighted avg       0.95      0.95      0.95     14307
```

Table III: Classification Report of LR

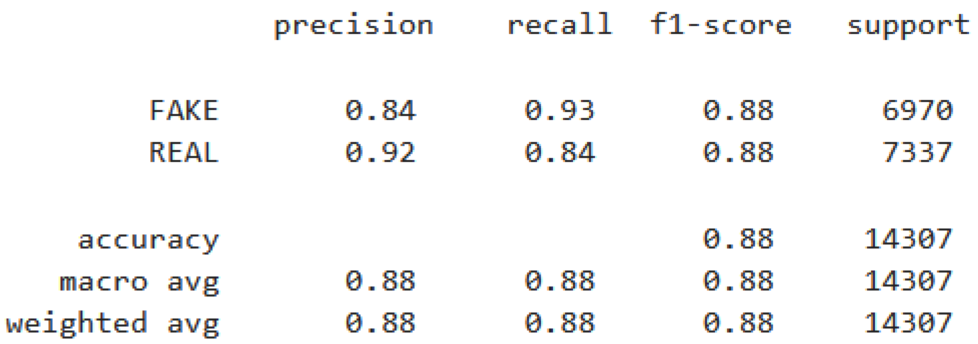


```
              precision    recall  f1-score   support

        FAKE       0.95      0.93      0.94      6970
        REAL       0.93      0.95      0.94      7337

    accuracy                           0.94     14307
   macro avg       0.94      0.94      0.94     14307
weighted avg       0.94      0.94      0.94     14307
```

Table IV: Classification Report of XGBoost

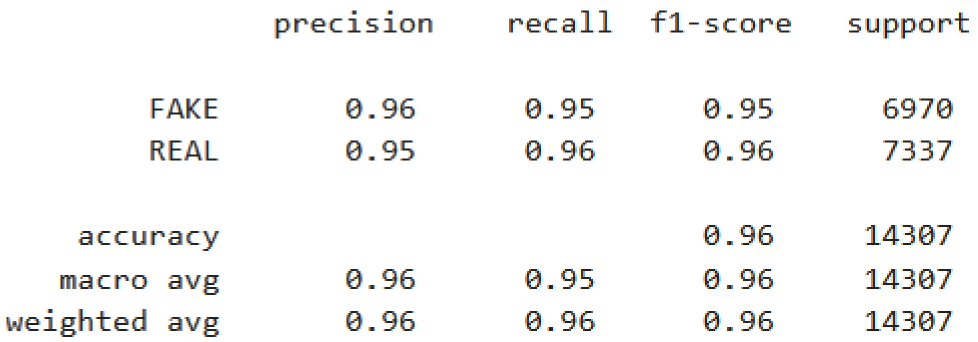


```
              precision    recall  f1-score   support

        FAKE       0.84      0.93      0.88      6970
        REAL       0.92      0.84      0.88      7337

    accuracy                           0.88     14307
   macro avg       0.88      0.88      0.88     14307
weighted avg       0.88      0.88      0.88     14307
```

Table V: Classification Report of Naïve Bayes

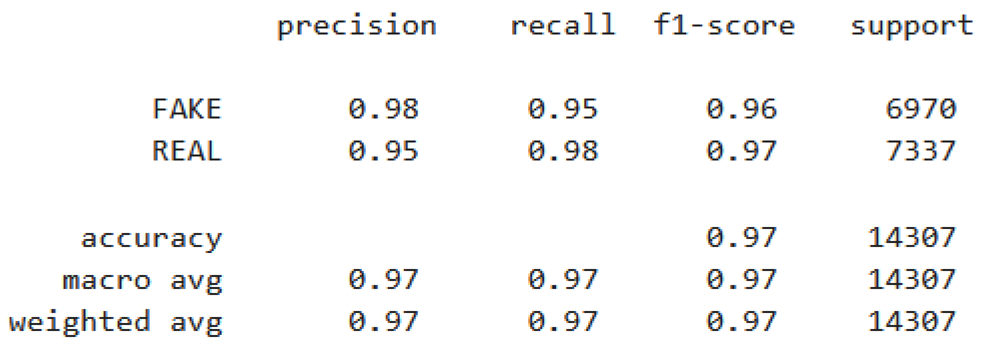


```
              precision    recall  f1-score   support

        FAKE       0.96      0.95      0.95      6970
        REAL       0.95      0.96      0.96      7337

    accuracy                           0.96     14307
   macro avg       0.96      0.95      0.96     14307
weighted avg       0.96      0.96      0.96     14307
```

Table VI: Classification Report of RNN

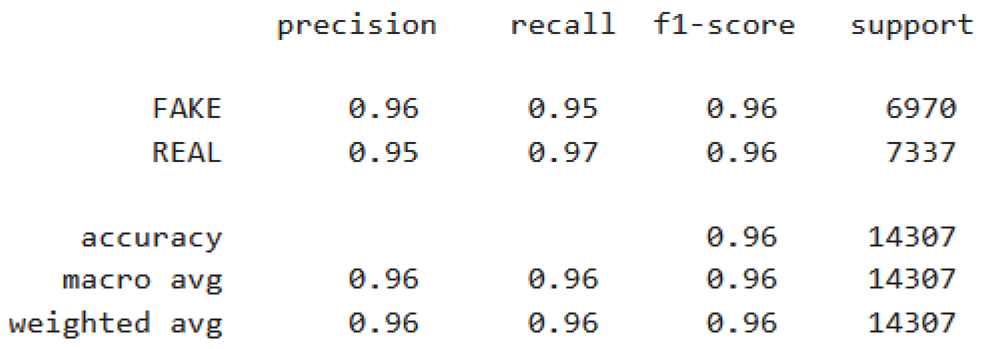


```
              precision    recall  f1-score   support

        FAKE       0.98      0.95      0.96      6970
        REAL       0.95      0.98      0.97      7337

    accuracy                           0.97     14307
   macro avg       0.97      0.97      0.97     14307
weighted avg       0.97      0.97      0.97     14307
```

Table VII: classification report of LSTM



```
              precision    recall  f1-score   support

        FAKE       0.96      0.95      0.96      6970
        REAL       0.95      0.97      0.96      7337

    accuracy                           0.96     14307
   macro avg       0.96      0.96      0.96     14307
weighted avg       0.96      0.96      0.96     14307
```

Table VIII: Classification Report of GRU

4.3 Overall Performance

Performance Metrics	**ML and DL Models**					
	LR	**XGB**	**NB**	**RNN**	**LSTM**	**GRU**
Accuracy	94.73%	95.38%	93.62%	95.51%	**96.58%**	95.70%
Precision	95%	94%	88%	96%	**97%**	96%
Recall	95%	94%	88%	95%	**97%**	96%
F1-score	95%	94%	88%	96%	**97%**	96%

Table IX: Overall Performance of all models

CHAPTER 5

CONCLUSION

In conclusion, this study of detection of faux news with the help of various ML and DL models showcases roles of traditional machine learning and advanced deep learning algorithms like XGBoost algorithm, RNN, LSTM, etc, in finding false claims or news on various platforms. This study found that LSTM had the best performance for the WELFake dataset with an accuracy of 96%. The results highlight how well LSTM architectures work to meet the challenges associated with detecting fake news. Their capacity to recognize reciprocal information flow and long-term dependency suggests that they may be able to identify subtle language patterns present in false material.

In terms of future work, we can focus on using more advanced algorithms like the transformer-based BERT or GPT for spotting fake information. We could also improve the current model by doing more extensive fine tuning with domain-specific knowledge.

REFERENCE

[1] Mitchell, T.M. and Tom, M. (1997) Machine Learning. McGraw-Hill, New York.

[2] Li Deng; Dong Yu, Deep Learning: Methods and Applications , now, 2014, doi: 10.1561/2000000039.

[3] W. J. Zhang, G. Yang, Y. Lin, C. Ji and M. M. Gupta, "On Definition of Deep Learning," *2018 World Automation Congress (WAC)*, Stevenson, WA, USA, 2018, pp. 1-5, doi: 10.23919/WAC.2018.8430387

[4] G. S. Mahara and S. Gangele, "Fake news detection: A RNN-LSTM, Bi-LSTM based deep learning approach," *2022 IEEE 1st International Conference on Data, Decision and Systems (ICDDS)*, Bangalore, India, 2022, pp. 01-06, doi: 10.1109/ICDDS56399.2022.10037403.

[5] U. P, A. Naik, S. Gurav, A. Kumar, C. S R and M. B S, "Fake News Detection Using Neural Network," 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS), Raichur, India, 2023, pp. 01-05, doi: 10.1109/ICICACS57338.2023.10100208.

[6] Shrutika S. Jadhav & Sudeep D. Thepade (2019) Fake News Identification and Classification Using DSSM and Improved Recurrent Neural Network Classifier, Applied Artificial Intelligence, 33:12, 1058-1068, DOI: 10.1080/08839514.2019.1661579

[7] Nasir JA, Khan OS, Varlamis I (2021) Fake news detection: a hybrid cnn-rnn based deep learning approach. Int J Inf Manag Data Insights 1(1):100007. https://doi.org/10.1016/j.jjimei.2020.100007

[8] P. Bahad, P. Saxena and R. Kamal, "Fake news detection using bi-directional LSTM-recurrent neural network", *Proc. Comput. Sci.*, vol. 165, pp. 74-82, Jan. 2019, [online] Available: http://www.sciencedirect.com/science/article/pii/S1877050920300806.

[9] real_or_fake, https://www.kaggle.com/rchitic17/real-or-fake, last accessed 2019/07/13.

[10] Fake News detection, https://www.kaggle.com/jruvika/fake-news-detection, last accessed 2019/07/13.

[11] M.-Y. Chen, Y.-W. Lai and J.-W. Lian, "Using deep learning models to detect fake news about COVID-19", *ACM Trans. Internet Technol.*, May 2022. https://doi.org/10.1145/3533431

[12] G. K. Shahi and D. Nandini. 2020. FakeCovid - A multilingual cross-domain fact check news dataset for COVID-19. arXiv: 2006.11343. Retrieved from https://arxiv.org/abs/2006.11343.

[13] COVID-19-News-Corpus. 2020. GitHub. Retrieved Sep. 15, 2020 from https://github.com/KangGu96/COVID-19News-Corpus/.

[14] Cofacts Open Datasets. 2020. Github. Retrieved Sep. 15, 2020 from https://github.com/cofacts/ opendata.

[15] A. Yerlekar, N. Mungale and S. Wazalwar, "A multinomial technique for detecting fake news using the Naive Bayes Classifier," 2021 International Conference on Computational Intelligence and Computing Applications (ICCICA), Nagpur, India, 2021, pp. 1-5, doi: 10.1109/ICCICA52458.2021.9697244.

[16] S. S. Reddy, S. Mandal, V. L. V. S. K. B Kasyap and A. R K, "A Novel Approach to Detect Fake News Using eXtreme Gradient Boosting," 2022 10th International Symposium on Digital Forensics and Security (ISDFS), Istanbul, Turkey, 2022, pp. 1-4, doi: 10.1109/ISDFS55398.2022.9800777.

[17] P. K. Verma, P. Agrawal, I. Amorim and R. Prodan, "WELFake: Word Embedding Over Linguistic Features for Fake News Detection," in IEEE Transactions on Computational Social Systems, vol. 8, no. 4, pp. 881-893, Aug. 2021, doi: 10.1109/TCSS.2021.3068519.

[18] V. Kanade, “What Is Logistic Regression? Equation, Assumptions, Types, and Best Practices”, https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/

[19] Miao Zou, Wu-Gui jiang, Qing-Hua Qin, Yu-Cheng Liu, Mao-Lin Lin, “Optimized XGBoost Model with Small Dataset for Predicting Relative Density of Ti-6Al-4V Parts Manufactured by Selective Laser Melting,” Materials 2022, 15(15), 5298; https://doi.org/10.3390/ma15155298

[20] A. Mehta, “Demystifying Naïve Bayes: Simple yet Powerful for Text Classification”, https://medium.com/@dancerworld60/demystifying-na%C3%AFve-bayes-simple-yet-powerful-for-text-classification-ad92b14a5c7

[21] M. Kaur and A. Mohta, "A Review of Deep Learning with Recurrent Neural Network," 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2019, pp. 460-465, doi: 10.1109/ICSSIT46314.2019.8987837

[22] R. Asghar, “Recurrent Neural Networks (RNNs) - Easily Explained”, https://dev.to/rimmel_codes/recurrent-neural-networks-rnns-easily-explained-5amm

[23] Gaikwad, R., Admuthe, L. (2024). Real-Time Sign Language Recognition of Words and Sentence Generation using MediaPipe and LSTM. In: Uddin, M.S., Bansal, J.C. (eds) Proceedings of International Joint Conference on Advances in Computational Intelligence. IJCACI 2022. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-97-0180-3_23

[24] S. Varshney, “Basics of Sequential Modelling , NLP and Large Language Models(LLM)”, https://www.linkedin.com/pulse/day-02-basics-sequential-modelling-nlp-large-language-varshney/

[25] Ivan Vasilev, Daniel Slater, Gianmario Spacagna, Peter Roelants, Valentino Zocca, “Python Deep Learning”, Second Edition, Packt Publishing, 2019

ANNEXURE-IV

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis A COMPARATIVE STUDY OF FAKE NEWS DETECTION USING MACHINE LEARNING AND DEEP LEARNING TECHNIQUES

Total Pages 51 Name of the Scholar NEVIL DOLPHY DSOUZA

Supervisor (s)

(1) MR. RAHUL

(2)

(3)

Department Software Engineering

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: TURNITIN Similarity Index: 11 %, Total Word Count: 8267

Date: 27/05/2024

NEVIL
Candidate's Signature

27/5/24
Signature of Supervisor(s)

PAPER NAME

Final Mtech Dissertation.pdf

WORD COUNT	CHARACTER COUNT
8267 Words	46951 Characters
PAGE COUNT	FILE SIZE
51 Pages	976.9KB
SUBMISSION DATE	REPORT DATE
May 27, 2024 2:41 PM GMT+5:30	May 27, 2024 2:42 PM GMT+5:30

● 11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 6% Internet database
- 3% Publications database
- Crossref database
- Crossref Posted Content database
- 10% Submitted Works database

● Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Small Matches (Less then 10 words)

● 11% Overall Similarity

Top sources found in the following databases:

- 6% Internet database
- 3% Publications database
- Crossref database
- Crossref Posted Content database
- 10% Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Delhi Technological University on 2024-05-23 Submitted works	3%
2	University of Wollongong on 2023-12-07 Submitted works	<1%
3	scholarworks.iupui.edu Internet	<1%
4	University of Hertfordshire on 2022-09-23 Submitted works	<1%
5	dspace.dtu.ac.in:8080 Internet	<1%
6	dspace.dtu.ac.in:8080 Internet	<1%
7	Leeds Beckett University on 2023-09-09 Submitted works	<1%
8	Asia Pacific University College of Technology and Innovation (UCTI) on... Submitted works	<1%

9	University of Kent at Canterbury on 2023-08-25 Submitted works	<1%
10	coursehero.com Internet	<1%
11	Phongsin Jirapipattanaporn, Worawat Lawanont. "Development of Ano... Crossref	<1%
12	mdpi-res.com Internet	<1%
13	Cranfield University on 2020-01-17 Submitted works	<1%
14	Liverpool John Moores University on 2022-06-04 Submitted works	<1%
15	University of Northumbria at Newcastle on 2022-01-20 Submitted works	<1%
16	Manipal Academy of Higher Education (MAHE) on 2023-08-02 Submitted works	<1%
17	ceur-ws.org Internet	<1%
18	tudr.thapar.edu:8080 Internet	<1%
19	Sheffield Hallam University on 2024-01-10 Submitted works	<1%
20	academic-accelerator.com Internet	<1%

21	ukmsarjana.ukm.my Internet	<1%
22	Asia Pacific University College of Technology and Innovation (UCTI) on... Submitted works	<1%
23	Universiti Tenaga Nasional on 2023-01-15 Submitted works	<1%
24	University of Bradford on 2022-11-25 Submitted works	<1%
25	University of Huddersfield on 2022-01-27 Submitted works	<1%
26	dspace.univ-tiaret.dz Internet	<1%
27	nemertes.library.upatras.gr Internet	<1%
28	towardsdatascience.com Internet	<1%
29	Coventry University on 2023-08-18 Submitted works	<1%
30	Indian Institute of Technology Goa on 2023-05-25 Submitted works	<1%
31	Liverpool John Moores University on 2022-08-30 Submitted works	<1%
32	Ravikumar, S.. "Machine learning approach for automated visual inspe... Crossref	<1%

33	University Politehnica of Bucharest on 2019-06-28 Submitted works	<1%
34	University of Hong Kong on 2023-01-24 Submitted works	<1%
35	University of Lincoln on 2017-04-27 Submitted works	<1%
36	University of Teesside on 2022-05-20 Submitted works	<1%
37	University of Wales Institute, Cardiff on 2023-03-29 Submitted works	<1%
38	Xiamen University on 2023-07-03 Submitted works	<1%
39	er.ucu.edu.ua:8080 Internet	<1%
40	lib.ptithcm.edu.vn Internet	<1%
41	pure.tue.nl Internet	<1%
42	svkm on 2024-05-10 Submitted works	<1%