

# FAKE NEWS DETECTION USING MACHINE LEARNING

TAN KIAN LONG

SESSION 2020/2021

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY

MULTIMEDIA UNIVERSITY

FEBRUARY 2021

# FAKE NEWS DETECTION USING MACHINE LEARNING

BY

TAN KIAN LONG

SESSION 2020/2021

THE PROJECT REPORT IS PREPARED FOR

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY  
MULTIMEDIA UNIVERSITY  
IN PARTIAL FULFILLMENT  
FOR

BACHELOR OF INFORMATION TECHNOLOGY  
B.I.T. (HONS) ARTIFICIAL INTELLIGENCE

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY

MULTIMEDIA UNIVERSITY

FEBRUARY 2021

© 2021 Universiti Telekom Sdn. Bhd. ALL RIGHTS RESERVED

Copyright of this report belongs to Universiti Telekom Sdn. Bhd as qualified by Regulation 7.2 (c) of the Multimedia University Intellectual Property and Commercialization policy. No part of this publication may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Universiti Telekom Sdn. Bhd. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

## DECLARATION

I hereby declare that the work has been done by myself and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

*Tan Kian Long*

---

Name of candidate Tan Kian Long

Faculty of Information Science & Technology  
Multimedia University

Date: 18/2/2021

## **ACKNOWLEDGEMENT**

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my supervisor, Prof. Dr. Lee Chin Poo for her invaluable advice, guidance and her enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parent and friends who had helped and given me encouragement

## **ABSTRACT**

Information and communication technology has evolved rapidly over the past decades, with a substantial development being the emergence of social media. It is the new norm that people share their information instantly and massively through the social media platforms. The downside of this is that the fake news also spread more rapidly and diffuse deeper than before. This has caused a devastating impact on people who are misled by the fake news. In the interest of mitigating this problem, fake news detection is crucial to help people differentiate the authenticity of the news. In this project, an enhanced convolutional neural network (CNN) model and Convolutional neural network and Long short-term memory (CLSTM) model are devised for fake news detection. The empirical studies on four datasets demonstrates that the two proposed models outshine the original models. Both of the proposed models have achieved a promising result and reduced the current overfitting problem.

## TABLE OF CONTENTS

<b>DECLARATION.....</b>	<b>III</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>IV</b>
<b>ABSTRACT.....</b>	<b>V</b>
<b>TABLE OF CONTENTS.....</b>	<b>VI</b>
<b>LIST OF TABLES .....</b>	<b>VIII</b>
<b>LIST OF FIGURES .....</b>	<b>IX</b>
<b>LIST OF ABBREVIATIONS/ SYMBOLS.....</b>	<b>X</b>
<b>LIST OF APPENDICES .....</b>	<b>XI</b>
<b>CHAPTER 1 INTRODUCTION .....</b>	<b>1</b>
1.1 Overview .....	1
1.2 Project Statement .....	2
1.3 Project Objectives .....	3
1.4 Project Scope.....	3
1.5 Report Organisation .....	4
<b>CHAPTER 2 LITERATURE REVIEW .....</b>	<b>5</b>
2.1 Overview .....	5
2.2 Existing fake news detection research .....	5
2.3 Feature extraction.....	13
2.4 Classification Algorithms .....	14
<b>CHAPTER 3 PROPOSED MODEL .....</b>	<b>22</b>
3.1 Overview .....	22
3.2 Process of the model .....	22
3.3 Pre-processing .....	23
3.4 Tokenize and padded sequence.....	23
3.5 Word Embedding .....	24
3.6 Model architectures .....	25

<b>CHAPTER 4 EXPERIMENT .....</b>	<b>33</b>
4.1 Overview .....	33
4.2 Datasets .....	33
4.3 Text Analysis .....	35
4.4 Experiment Setup .....	35
4.5 Evaluation Method .....	40
4.6 Result .....	41
<b>CHAPTER 5 CONCLUSION .....</b>	<b>51</b>
5.1 Conclusion .....	51
5.2 Future work .....	52
<b>REFERENCES .....</b>	<b>53</b>
<b>APPENDICES .....</b>	<b>55</b>



## LIST OF TABLES

<b>Table 2.1:Summary table of Existing Fake News Detection Methods .....</b>	<b>11</b>
<b>Table 4. 1:Comparison of title and text length in different dataset .....</b>	<b>35</b>
<b>Table 4. 2: Performance of each method on Dataset1 .....</b>	<b>42</b>
<b>Table 4. 3: Performance of each method on Dataset2 .....</b>	<b>43</b>
<b>Table 4. 4: Performance of each method on Dataset3 .....</b>	<b>44</b>
<b>Table 4. 5: Performance of each method on Dataset4 .....</b>	<b>45</b>
<b>Table 4. 6: Differences of original CNN and modified CNN.....</b>	<b>46</b>
<b>Table 4. 7: Differences of the original and modified CLSTM model .....</b>	<b>47</b>

## LIST OF FIGURES

<b>Figure 1. 1: Phase 1 Gantt Chart</b> .....	4
<b>Figure 1. 2: Phase 2 Gantt Chart</b> .....	4
<b>Figure 2. 1: example of decision tree</b> .....	16
<b>Figure 2. 2: Prediction in Random Forest</b> .....	17
<b>Figure 2. 3: SVM example</b> .....	17
<b>Figure 2. 4: Sigmoid function graph</b> .....	18
<b>Figure 2. 5: CNN used in image task</b> .....	19
<b>Figure 2. 6: structure of LSTM in RNN</b> .....	20
<b>Figure 2. 7: Bert input representation</b> .....	21
<b>Figure 3. 1: Process of the proposed model</b> .....	22
<b>Figure 3. 2: example of tokenized sentence and word index</b> .....	24
<b>Figure 3. 3: Example of word embedding in geometric space</b> .....	25
<b>Figure 3. 4: CNN architecture</b> .....	26
<b>Figure 3. 5: example of word embedding</b> .....	26
<b>Figure 3. 6: CNN model structure</b> .....	29
<b>Figure 3. 7: CLSTM architecture</b> .....	29
<b>Figure 3. 8: CLSTM model structure</b> .....	32
<b>Figure 4. 1: Before and after of text pre-processing</b> .....	36
<b>Figure 4. 2: Part of the TF-IDF matrix</b> .....	36
<b>Figure 4. 3: CNN Architecture</b> .....	38
<b>Figure 4. 4: LSTM and bidirectional LSTM Architecture</b> .....	39
<b>Figure 4. 5: CLSTM Architecture</b> .....	40
<b>Figure 4. 6: Plots of the loss (CNN)</b> .....	49
<b>Figure 4. 7: Plots of the loss (CLSTM)</b> .....	50

## **LIST OF ABBREVIATIONS/ SYMBOL**

FIST	Faculty of Information Science and Technology
FYP	Final Year Project
MMU	Multimedia University
SVM	Support Vector Machine
RF	Random Forest
LR	Logistic Regression
DT	Decision Tree
PA	Passive Aggressive
NB	Naïve Bayes
CNN	Convolutional Neural Network
MCNN	Modified Convolutional Neural Network
LSTM	Long Short-Term Memory
CLSTM	Convolutional Neural Network + Long Short-Term Memory
MCLSTM	Modified Convolutional Neural Network + Long Short-Term Memory
Bi-LSTM	Bidirectional - Long Short-Term Memory
Bert	Bidirectional Encoder Representations from Transformers

## **LIST OF APPENDICES**

<b>Appendix A: Meeting logs .....</b>	<b>1</b>
<b>Appendix B: Checklist .....</b>	<b>7</b>

# CHAPTER 1

## INTRODUCTION

### 1.1 Overview

In these few couple years, the online news and social media has gradually become the main sources for people to receive information because the rapid improvement of internet and technology. For example, people can acquire news from Facebook, Twitter, or any news website with just one click. However, most people normally do not know which pieces of information are correct or incorrect. Therefore, detecting the fake or rumour news has become the main concern nowadays. It is because the spread of fake news may bring problems to the people who are misled by it.

There are several types of definitions for the term fake news. Generally, fake news means deceptive news or propaganda that is used to mislead people or even affect people's decisions and options. Fake news is created to gain benefits for reputation, financial or political. For example, there is some fake review that appears on the internet intent to pursue the consumer to buy their product. If the consumer does not have the basic knowledge of the product, in the end, the consumer may believe the fake review. Hence, fake news detection is needed because most people who do not have the basic knowledge of the information or insufficient time to check the credibility of the news may believe in the news even if it is fake news.

Besides, some organizations have developed the detection tools for fake news, for example, politifact.com and snopes.com have their own tools to detect the level of fake news. Nevertheless, these tools need manual work so it is costly and time-consuming. In recent years, several challenges of tasks are organized to deal with fake news, including Fake news challenge stage 1(FNC-1), Web Search, and Data Mining (WSDM) fake news challenge, clickbait-challenge and etc.

From the previous work, traditional machine learning methods and deep neural networks have been applied to detect fake news. Some of the researchers collect the dataset themselves to develop the model. This project will use the public dataset that is available online to train the model. Fake news appears in a variety of forms, such as text, video, audio, and image. For example, Yang et al. (2018) did the research based on text and image using Convolutional Neural Network. In this project, it will only focus on text-based fake news.

## **1.2 Project Statement**

There are many existing fake news detection methods have been developed by people nowadays. It is because the recent appearance of the technology like the technique of Artificial Intelligence (AI), Machine learning and Natural Language Processing (NLP). These techniques have attracted the passionate researchers to try these methods on their experiment in order to solve the epidemic of fake news on social media. The researchers are using different methods to deal with the data text and train the models. Therefore, a study is required to understand the process and how these fake news detection methods work on fake news detection.

Although there are a lot of fake news detection methods available today, the performance of the methods is still not considered as good method. Some of the deep learning methods which are used in detecting fake news also faced the overfitting problem. An overtrained model will not get good result in testing. On today's life, fake news has spread rapidly on social media and has brought a devastating problem to human simultaneously. To prevent human affect by the fake news, a good and effective model is required. In view of these, some enhancement is essential to mitigate these problems and improve the models.

Fake news detection is always a hot topic for people. Recently, the researchers and some of the organisation has collected the datasets themselves and organized competition to let people to evaluate the performance of their methods. Hence, there are many fake news datasets available online. After the enhancement being made in the model, some experiments are needed to be conducted to evaluate the performance

of the fake news detection methods. Thus, result of the models can proof that the enhancements are effective to solve the problem.

### **1.3 Project Objectives**

This project aims to study the existing fake news detection algorithm and enhance the algorithm to get better performance so that the fake news destructive problem can be reduced and help the public not to be affected by fake news.

The objective of this research includes as shown below:

- To study the existing classification methods
- To enhance the existing models for fake news detection
- To evaluate the performance of the enhanced models

### **1.4 Project Scope**

In order more understand about the fake news detection and how the other research done their experiment, the research papers will be compiled into literature review. Besides that, comparison among the existing method also has been done to understand which methods perform better in fake news classification. After understanding the limitation of the existing method, the proposed model will be proposed to improve the performance and get better result. The process of the proposed model also will be explained in the report.

#### **1.4.1 Gantt Chart**

Based on the figure 1.1, phase 1 will start the project from project registration. After choosing and confirming the title, studying the past research of fake news detection will be the next step. The knowledge and the method of fake news detection need to compile into the literature review. Then, the experiment will be started by using the public dataset to test in the existing method. The result of the experiment will be added to the report. The last step is to submit the finalized report and do the presentation.

Task Name	Start date	Finish date	Duration	July				August				September				October			
				W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Project Registration	30/7/2020	30/7/2020	1day																
Study Past Research(Literature review)	1/8/2020	14/8/2020	2weeks																
Try the datasets on existing method	8/8/2020	22/8/2020	2weeks																
enhance proposed model	22/8/2020	12/9/2020	3weeks																
Compile the result into report	12/9/2020	26/9/2020	1weeks																
Finalize phase 1 report	26/9/2020	1/10/2020	5days																
Presentation	5/10/2020	5/10/2020	1day																

**Figure 1. 1: Phase 1 Gantt Chart**

According to figure 1.2, phase 2 will continue the project with adding some new literature review to summarize more technique from other researchers. Then, there is another dataset added into the experiment to train and test the models. After that, the proposed model 2 will also be added into the report. After compiling all the result into the report, the report for phase 2 will be finalized. Next, the proposed model will be written into conference paper to publish. The phase 2 will be ended with presentation.

Task Name	Start date	Finish date	Duration	December				January				February			
				W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Add literature review	7/12/2020	14/12/2020	1week												
Add one more the datasets	14/12/2020	21/12/2020	1week												
enhance proposed model 2	21/12/2020	4/1/2021	2weeks												
Compile the result into report	4/1/2021	11/1/2021	1week												
Finalize phase 2 report	11/1/2021	18/1/2021	1week												
write proposed model into conference paper	18/1/2021	1/2/2021	2weeks												
Presentation	18/2/2021	18/2/2021	1day												

**Figure 1. 2: Phase 2 Gantt Chart**

## 1.5 Report Organisation

This report contains five chapters. Chapter 1 will explain how the fake news problem has affected other life and the reason why fake news detection is needed. The problem statement and objective of this project will also be included in this chapter. The techniques and the experiment result of fake news detection from other researchers will be compiled as literature reviews to present in Chapter 2. Besides, the explanation of the existing classification methods will be shown in the same chapter. There is a summary table of the literature reviews that clearly stated that what contribution has been done so far in this domain. In Chapter 3, the process and architecture of the proposed models will be described. In Chapter 4, the experiment result will be shown. Finally, the conclusion and future work of fake news detection will be discussed in Chapter 5.



## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Overview**

Chapter two will discuss and explain the research that is done by other researchers in the fake news detection domain. Section 2.2 is about the techniques that summarize from the paper. Section 2.3 will explain the classification algorithms that can be used to detect fake news.

#### **2.2 Existing fake news detection research**

Based on the paper, Gilda, (2018) cleaned and eliminated the duplicated articles from the datasets before splitting the datasets for training and testing. The models are made in three combinations to compare the performance based on the result. The combinations are Term Frequency-Inverse Document Frequency (TF-IDF) bigram with five classification methods, Probabilistic Context-Free Grammar (PCFG) bigram with five classification methods, and both TF-IDF and PCFG bigram with five classification methods. The five classification methods namely Support Vector Machine (SVM), Stochastic Gradient Descent, Gradient Boosting, Bounded Decision Tree, and Random Forests. Eventually, the best performance model is TF-IDF with Stochastic Gradient Descent. The model has achieved 77.2% accuracy in the experiment.

In this research paper, Girgis & Gadallah, (2018) proposed a model using 3 types of deep learning models including Vanilla, Long Short-Term Memory (LSTM) and Gated recurrent units (GRU). They used the LIAR dataset in their experiment. In their proposed model, they did some steps to pre-processing the data. Each sentence was split to deal with separately. Then, the stop words (such as the, an, a, etc) are removed because these types of words normally are meaningless. Using word embedding to analyse the relationship of the words. For example, "see" and "watch" are syntactically different, but their meanings are somehow related. After that, using the deep learning algorithms to classify the text whether it is real or fake. The final

result has shown up that the GRU model got the best result among the three because it is a free memory unit, easy to modify, and solved the gradient vanishing problem that occurs in the Vanilla model.

According to the research done by Agarwalla et al. (2019), they built three types of machine learning models, namely naive Bayes with Lidstone smoothing, support vector machine and logistic regression to classify the news whether it is fake or true. They collect 2136 of fake news articles and 1872 of the real news articles. In the pre-processing part, the authors used the NLTK package to remove the stop-words in headline text and body text. The authors used the word clouds to analyse the frequency and importance of the words that appear in fake or true news. In the split training and testing part, the datasets were split to 70% of data for training whereas 30% of data will be used for testing. Finally, they found out that the most promising result from the experiment was the body and headline text that fed on naive Bayes with Lidstone smoothing. The accuracy reached 83.16 percent.

According to the work (Granik & Mesyura, 2017), they used a simple naive Bayes algorithm to classify the news articles. In the experiment, the author removed the article with the “null” text and only considered the label with "fake" and "true" before feeding into the classifier. The paper also discussed some ways to improve the performance the classifier, such as increase the quantity of the datasets, get the article with longer text, remove stop-words, using stemming to treat the similar words in the text and group the words in the text so that it can help to understand the syntax construction in the text.

Paper done by Jwa et al. (2019) have enhanced the Bidirectional Encoder Representations from Transformers (BERT) to form a BAKE model. The BAKE model mitigates the data imbalance problem by using weighted cross entropy (WCE) to categorize the data. The experiments were separated into two parts, namely pre-training and task-specific fine-tuning. In the pre-training, the BERT model was trained using Wikipedia and Book Corpus datasets. In the model fine-tuning, the Fake News Challenge (FNC-1) dataset was used. The weighted cross entropy (WCE) was used to classify the dataset. The authors further combined extra unlabelled news corpora,

which included the CNN news from United Kingdom and the Daily Mail news from United States in order to form the exBAKE model. Finally, the exBAKE model achieved a F-1 score of 74.6%

Ozbay & Alatas (2020) used twenty-three supervised learning methods to classify the fake news datasets that were collected from the real-world. The authors used three different datasets to train and test the models. They extracted the body articles and the label from the datasets to run in experiment. The model starts with data pre-processing. It included tokenization, stop-words removal, and stemming. Then, they used term frequency and created a Document-Term Matrix as feature extraction for the model. It then fed into the classification model. The experiment used value of accuracy, precision, recall and F-measure to evaluate the model. According to the result, Decision Tree, ZeroR, CVPS, and WIHW algorithm get a better accuracy to compare with others.

From research done by Ahmed et al. (2017), they used two types of common feature extraction techniques which includes Term Frequency(TF) and Term Frequency -Inverted Document Frequency(TF-IDF) and 6 different classification algorithms such as K-Nearest Neighbour (KNN), Stochastic Gradient Descent (SGD), Support Vector Machines (SVM), Linear Support Vector Machines (LSVM), Logistic Regression(LR) and Decision Trees (DT)) to compare the performance. The size of the n-gram will be tried from 1 to 4. The dataset is compiled from two website which is Reuters and Kaggle. Stop-words removal and Stemming are used in the data pre-processing part. Then, they used different types of feature extraction to feed the data into different classifiers to compare the result. In the splitting process, 80% of data text was used for training and 20% of the data text was used for testing. The cross-validation will value of 5 also will be included in this phase. The best accuracy among the models is LSVM with TF-IDF of unigram size. It reaches 92% accuracy.

Khan et al. (2019) built traditional Machine learning include Support Vector Machine, Decision Tree, Logistic Regression, Naive Bayes, Adasboost, K-neighbours classifier(K-NN)) and neural network models include CNN, LSTM, Bi-LSTM, C-LSTM, HAN, Convolutional HAN, and Character-level C-LSTM to analysis the

performance and compare the result. There are three types of datasets being used in the experiment such as Liar, Fake or Real News, and Combined Corpus. During the dataset pre-processing part, they cleaned the raw text by eliminating the IP and URL addresses. Then, stop-words removal and correction of the spelling words have also been done. Using the NLTK library of Snowball Stemmer to perform stemming. In the experiment, Lexical and Sentiment features extraction and n-gram features extraction are used in each model. In the neural network model, they used the pretrained weight of GloVe to represent the word vector in the embedding layer instead of using random weight. Comparing all traditional method result from the experiment, Naive Bayes with n-gram (bigram TF-IDF) uses a combined corpus as the dataset gets the best result. It reached 94% accuracy. Among the neural network models, no model is outstanding than others. In neural network models, the author indicated that these models suffered from overfitting problems while using LIAR datasets.

Based on the research(Bahad et al., 2019), they compared the performance of methods including CNN, vanilla RNN unidirectional LSTM, and Bi-directional LSTM. Removal of stop-words and punctuation and UTF-8 format was used during text pre-processing. Global Vectors for Word Representation (GloVe) embeddings are also applied to analyse how the headline in the article and the main content of the article will affect the result. The data are separated into 60% for training,20% for validation, and 20% for testing in each method equally. The hyperparameters like AdaGrad, RMSProp, and Adam are used in these methods to get a better result. Eventually, Bi-directional LSTM-RNN performs the best accuracy. According to the author, the methods also have their own superior like CNN is good at extracting the local feature of the data whereas LSTM-RNN is expert in solving the sequence data like text, movie and etc.

Kaur et al. (2020) proposed a model named multi-level voting. The experiment has included three feature extraction techniques like Count-Vectorizer (CV), Hashing-Vectorizer(HV) and Term Frequency-Inverse Document Frequency (TF-IDF). Besides that, there are twelve classifiers including Passive Aggressive (PA), Support Vector Machine, (SVC), Multilayer Perceptron (MLP), NuSVC, Decision Tree (CART),, Stochastic Gradient Descent(SGD), LinearSVC, Naïve Bayes(NB), Logistic

Regression (LR), AdaBoost, Gradient Boosting and Voting. There are three types of datasets used in this model to evaluate the performance. There are few steps need to be done during pre-processing, such as remove the redundant data, non-title data, stop words, null values of text are used to clean the noise in the text. The data are separated into a ratio of 0.67 (training): 0.33(testing). In the classification phase, it has separated into three-levels. According to the result, the multilevel voting model gets the best accuracy and efficiency compared to others.

Research was done by Thakur et al. (2020) which was using Convolutional Neural Network and Gradient Boosted Decision Tree to classify the news as real or fake. TF-IDF Count, SVD, Sentiment and Word2V are used in the experiment as feature extraction. The proposed model has two important components, news extractor, and stance detection. The two popular extractor API, Microsoft Azure Natural Language Processing API and IBM Watson Natural Language Understanding API are used as keyword extractors to extract the given keyword of the headline or text from the database. Two of the classification methods are going to be used in Stance detection to detect whether the given text is fake or real news. After comparing the result, the combination of the two methods gets the best accuracy that is 97.39%.

Hiramath and Deshpande (2019) used five types of machine learning classifiers to make comparison and lastly proposed a model that gets the best performance. It included Naïve Bayes (NB), Support vector machine (SVM), Logistic regression (LR), Random Forest (RF), and deep neural network (DNN). Dataset is crawled by themselves using the Java system as the framework and MySQL as the backend to set up the experiment. Two methods are used in pre-processing part in order to clean the text. First step is to remove the stop words so that it can reduce the meaningless words. Second step is the stemming application which can change the word to the root word so that there will be less ambiguity during training. In the end, DNN gets the best accuracy and requires less time to perform the processing in the experiment.

According to the research, Poddar et al. (2019) were using 2 different feature extraction including Term Frequency-Inverse Document Frequency(TF-IDF) and Count Vectorizer(CV) in each different type of machine learning classifier. The

classifiers are Naive Bayes, SVM, Logistic Regression, Decision Tree, and Artificial Neural Network. The dataset came from Kaggle that the contents are associated with BS Detector Chrome Extension by Daniel Sieradski. Stop words removal is used in the text pre-processing. SVM with TF-IDF vectorizer gets the highest accuracy, it is 92.8%.

Amine et al. (2019) did research using different characteristics from the dataset to feed into the Convolutional Neural Network to test which types of data can achieved a good result in the experiment. The dataset is from the Kaggle website. Lower casing, punctuation, and stop-words removal, stemming, tokenization, and padding are applied to prepare text according to the following sequence. In the experiment, they used 90% of data to train the model and retained 10% to test the model performance.. According to the paper, the model used 50 training epochs to perform classification. The characteristic of text combined with the author gets the highest accuracy. It achieved 96% accuracy.

From the research(Benamira et al., 2019), they proposed graph-based semi-supervised learning to detect fake news articles. The authors compared the proposed model with other traditional machine learning methods. The model used GloVe word embedding to calculate the appear word frequency. Then, using the graph construction to observe the similarity of the article. Lastly, it fed into the classifiers: Graph Convolutional Networks (GCN) and Attention Graph Neural Network (AGNN).

Kesarwani et al. (2020) was using K-Nearest-Neighbour methods as the classifier to deal with fake news detection. In the experiment, they used the dataset from buzz feed news which collected from the Facebook post. The news articles are labelled into four types including "mostly true", "mostly false", " mixture of true and false" and " no factual content". The total articles are 2282. It included videos, images, URLs, and texts. In the train-test part, 80% of the data has been fed into training phase. While training model, the authors tried different value K to get the best performance. According to this paper, the model has reached 79% of accuracy, 75% of average precision and 79% of recall eventually. The authors also figured out that the model will get the best performance while the value of k in the classifier is around 15 to 20.

Based on the paper research, there are six methods used to detect the fake news including NB, DT, CNN, CNN-LSTM, KNN and RF(Kaliyar, 2018). The author collected the dataset from kaggle.com and Signal Media website. The combination of dataset contains title, id, author and label. The data are labelled in fake and real. In the experiment, the author used two types of feature extraction to deal with the text data which include TF-IDF and Hashing vectorizer. From the result shown in the paper, the very deep CNN performs the best result which 98.3% accuracy.

**Table 2.1:Summary table of Existing Fake News Detection Methods**

Paper Authors	Feature	Classification methods	Datasets
(Gilda, 2018)	TF-IDF, PCFG	SVM, Stochastic Gradient Descent, Gradient Boosting, Bounded Decision Tree and Random Forests	Dataset from Signal Media
(Girgis & Gadallah, 2018)	Word embedding	SVM, LR, Bi-LSTM, CNN, Vanilla, GRU, and LSTM	LIAR dataset
(Agarwalla et al., 2019)	-	Naive Bayes with Lid stone smoothing, logistic regression, SVM	Taken from Kaggle contains URLs, Headline, Body, and Label
(Granik & Mesyura, 2017)	-	Naive Bayes	Collect from BuzzFeed news
(Jwa et al., 2019)	-	BERT	FNC-1 dataset
(Ozbay & Alatas, 2020)	TF, Document Term Matrix	BayesNet, JRip, OneR, Decision Stump, ZeroR, SGD, CVPS, RFC, LMT, LWL, CVC, WIHW, Ridor, MLP, OLM, Simple Cart,	ISOT Fake News dataset, Random Political News dataset,

		ASC, J48, SMO, Bagging, Decision Tree, IBk, and KLR	BuzzFeed Political News dataset
(Ahmed et al., 2017)	TF, TF-IDF, n-gram	SGD, SVM, LSVM, KNN DT	compiled from Reuters.com and kaggle.com
(Khan et al., 2019)	Lexical and Sentiment, n-gram	SVM, Logistic Regression, Decision Tree, Naive Bayes, K-NN, Adaboost, CNN, LSTM, Bi-LSTM, C-LSTM, HAN, Convolutional HAN, and Character-level C-LSTM	LIAR, Fake or Real News, and Combined Corpus
(Bahad et al., 2019)	GloVe (word embedding)	CNN, vanilla RNN unidirectional LSTM, and Bi-directional LSTM.	Two datasets from Kaggle.com
(Kaur et al., 2020)	TF-IDF, CV, HV	NB, SVC, NuSVC, LinearSVC, Decision Tree (CART), PA, SGD, LR, MLP, AdaBoost, Gradient Boosting and Voting	News trends, Kaggle, Reuters
(Thakur et al., 2020)	SVD, Count, Sentiment, TF-IDF, and Word2V	Gradient Boosted Decision Tree and Convolutional Neural Network	Taken from Kaggle contains URLs, Headline, Body, and Label
(Hiramath & Deshpande, 2019)	-	LR, NB, SVM, RF, DNN	Own dataset
(Poddar et al., 2019)	CV, TF-IDF	NB, SVM, LR, Decision Tree, NN	Kaggle dataset



(Amine et al., 2019)	Tokenization	Convolutional Neural Network	Kaggle datasets
(Benamira et al., 2019)	GloVe word embedding	Graph Convolutional Networks (GCN) and Attention Graph Neural Network (AGNN)	Public dataset
(Kesarwani et al., 2020)	-	K-Nearest Neighbor Classifier	Buzz Feed News (Facebook Post)
(Kaliyar, 2018)	TF-IDF, CV, HV	NB, DT, CNN, CNN-LSTM, KNN, RF	Kaggle and Signal Media

### 2.3 Feature extraction

Fake news detection belongs to one kind of text classification. In the text classification, there are a huge number of phrases, terms, and words involved in the learning process. To avoid the high computational burden, feature extraction is needed in the process to reduce the irrelevant or redundant features so that the size of the feature space dimension will be reduced. These will directly affect the accuracy of the classifier. In this subtopic, the Term Frequency-Inverse Document Frequency (TF-IDF), which is one of the well-known feature extractors used in text classification will be explained.

#### 2.3.1 TF-IDF

TF-IDF is one of the text feature extraction which can use to understand the importance of the key words in the document. There are two part of calculation. First is the TF part. It refers to how often the word shows up in the particular document. In order to get the word frequency of each document in mathematical calculation, the number times of the word ( $n_{i,j}$ ) which shows up in the document divided by the total words in the document ( $\sum_k n_{i,j}$ ). The term frequency formula is as below:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}} \quad (2.1)$$

Normally, the words with high frequency are the meaningless words like is, are, a, an and the, etc. Then, Inverse Document Frequency is to calculate how often or seldom the words in the document. Equation 2.2 is the IDF mathematical formula. N in the formula refers to the total number of document and  $df_i$  is the number of the document which contains the specific words. If the words are often show up in the document, the value will near to 0 or equal to 0.

$$idf(w) = \log \frac{N}{df_t} \quad (2.2)$$

Then, multiplying TF and IDF together will get the TF-IDF score of the word in the document. The higher value of the score, the more meaningful the word in the corresponding document. Equation 2.3 is the mathematical formula. As below,

$$w_{i,j} = tf_{i,j} * \log \frac{N}{df_i} \quad (2.3)$$

The advantages of this algorithm are the simplicity, high speed processing and easy to understand. However, using the term frequency to find the keywords of the document is not comprehensive because sometimes the keywords wouldn't appear frequently. Besides, this algorithm also difficult to show the relationship of the words with closest meaning.

## 2.4 Classification Algorithms

Nowadays, there are a lot of machine learning algorithms have been applied to detect and solve the fake news disaster. In this subtopic, nine classification algorithms will be explained. For the traditional classification algorithms, it includes Naive Bayes, Passive Aggressive, Decision Tree, Random Forest, SVM, Logistic regression. Deep learning algorithms are such as CNN, LSTM, and BERT.

### 2.4.1 Naïve Bayes

The research from Granik & Mesyura, (2017) found that Naive Bayes is a good classifier to solve problems like fake news detection although it is a simple approach. It is a kind of probabilistic machine learning algorithm (supervised learning) based on Bayes Theorem. The mathematical formula is shown in 2.4. From 2.4 equation, x, y means the events for example it is fake news or real news. P(x) and P(y) refer to the probability of event x or y occurs in the situation. Therefore, P(x|y) means to find the probability of x under the event y whereas P(y|x) means to get the probability of y under the situation x.

$$P(x|y) = \frac{P(y|x).P(x)}{P(y)} \quad (2.4)$$

In the coding part, Naive Bayes has few types of algorithms such as Gaussian, Multinomial, and Bernoulli. However, Multinomial Naive Bayes is the one which best deals with text classification problem. Naive Bayes algorithm is also good at the fast prediction of the test data set. It can perform classification well in the condition that the assumption of independence holds. However, the limitation is also very obvious because it is nearly impossible to get independent predictors in real life.

### 2.4.2 Passive Aggressive

Passive Aggressive algorithms are one of the machines learning algorithms. It is usually used to solve large-scale training, especially for the online input data, for example, used to detect fake news on Twitter and other social media because it can deal with the big amount of data that is being added at a fast pace. Passive and Aggressive have different meanings. Passive means the model will remain the same if the prediction is correct whereas Aggressive means the model will change if it predicts wrongly. It is a bit similar to the perceptron model. It only includes the regularization parameter but does not use the learning rate. There are some important parameters used in this algorithm such as regularization parameter(C), Maximum number of iterations (max\_iter), and stopping criterion(tol).

### 2.4.3 Decision Tree

Decision Tree is normally applied for solving the regression and classification problem. It is using the tree structure concept to apply the algorithm. Each of the leaf nodes represents the class label and the branches are the features that can lead to the class. It will start at the root node and compare the features in each of the internal nodes and get the final answer. Figure 2.1 shows how a simple decision tree classify the following task as an example.

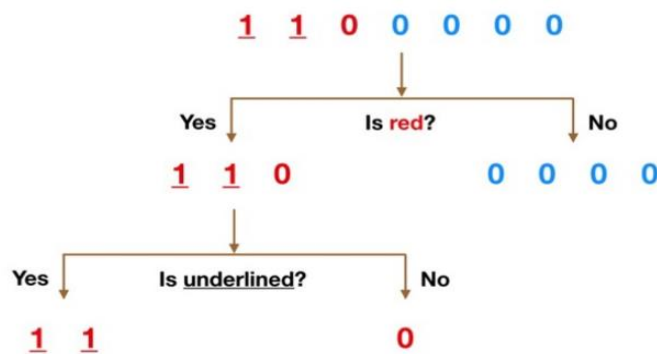
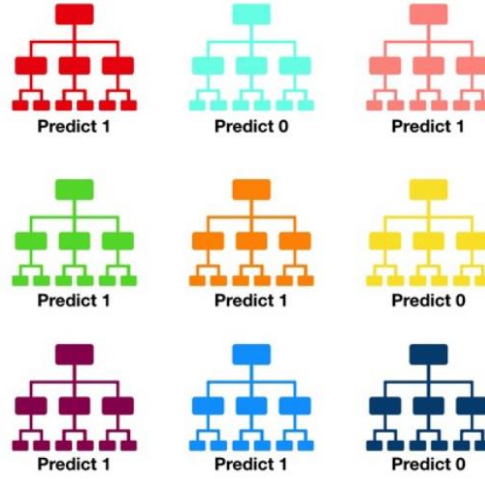


Figure 2. 1: example of decision tree

### 2.4.4 Random Forest

Random Forests also are known as Random decision forests. It is a types ensemble learning which is often used by data scientists to do prediction and classification. It actually contains a huge amount of individual decision trees to perform a task. Each individual decision tree is separated into a class prediction from the random forests. Eventually, the class that contains the majority will become the prediction model, for instance, in case figure 2.2, the prediction model is 1. It used two methods to ensure uncorrelated forest of tree in the model, including Bootstrap Aggregation and Feature Randomness.

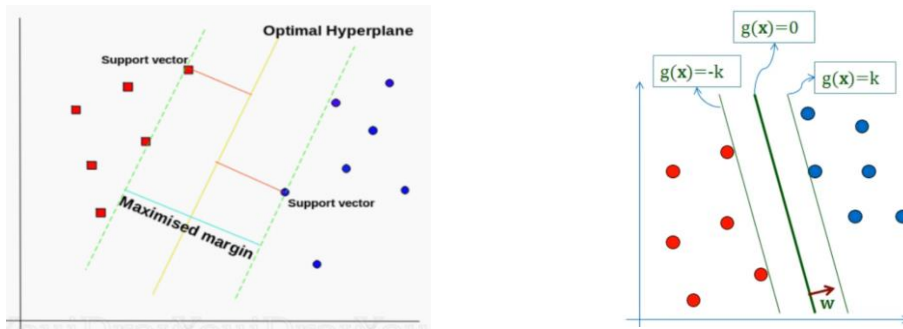


**Figure 2. 2: Prediction in Random Forest**

#### 2.4.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) is broadly applied in the categorisation task due to its effectiveness and powerfulness of this algorithm. Agarwalla et al. (2019) used SVM to do the research. The basic concept of SVM is it generates a hyperplane in the dimensional space to separate the training data from two different classes. In the SVM algorithm, the closest point of both classes to the hyperplane is called support vectors. Then, the margin is the distance between the two support vectors from both classes. Figure 2.3 shows the components of the SVM algorithm. 2.5 shows the equation of the distance between support vector and optimal hyperplane, where  $g(x)=0$  is hyperplane and  $w$  refers to margin.

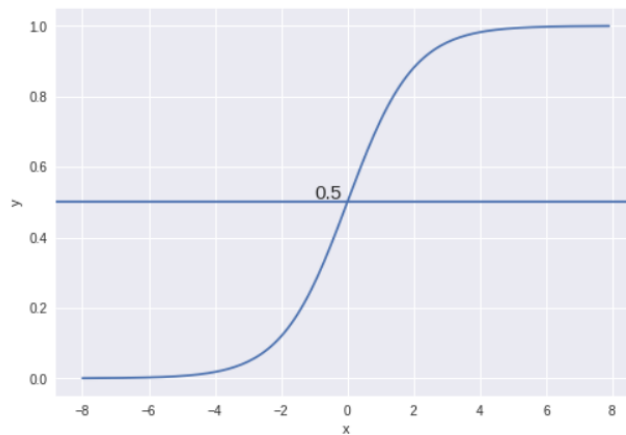
$$g(x) = w^T x_i + b \quad (2. 5)$$



**Figure 2. 3: SVM example**

### 2.4.6 Logistic Regression

This method is a kind of machine learning algorithms which is applied to classify the discrete set of classes. Normally, there are two types logistic regression including binary and Multi-linear function. Binary means that there are two labelling such as the fake news detection which detect whether the news is true or fake. Whereas, multi-linear function means there are many labelling in the data, for example, recognize types of vehicle which including van, car, bus, motorcycle and etc. This algorithm is using the idea of probability to predict the case. The cost function in Logistic Regression is called as Sigmoid function. The value of this function will be set between 0 to 1. Below figure 2.4 is the graph that show the Sigmoid function. For example, given a situation fake news-0 and true news-1. Then, set a threshold to 0.5. With this setting, it can classify the news which the value above 0.5 will be the true news and below 0.5 will be fake news. Equation 2.6 is the mathematical equation of sigmoid function and equation 2.7 is the logistic regression hypothesis.



**Figure 2. 4: Sigmoid function graph**

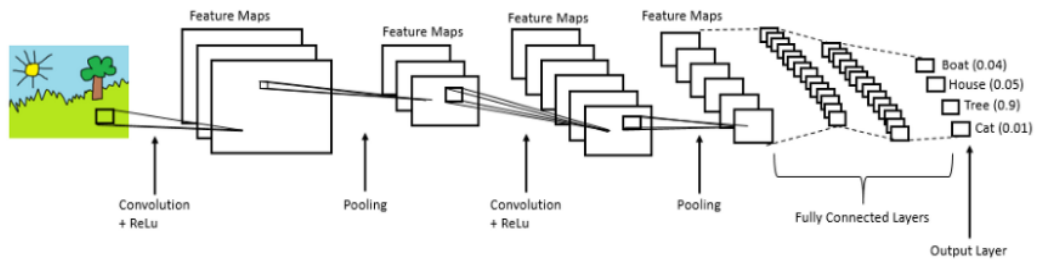
$$f(x) = \frac{1}{1 + e^{-(x)}} \quad (2. 6)$$

As above equation,  $f(x)$  refers to the sigmoid function and  $e$  is the Euler's number. The logistic regression formula is calculating the probability where  $\beta_0$  and  $\beta_1$  are the coefficient and  $x$  refers to the independent variable.

$$P = \frac{e^{(\beta_0 + \beta_1 X)}}{1 + e^{(\beta_0 + \beta_1 X)}} \quad (2.7)$$

### 2.4.7 CNN

Convolutional Neural Network (CNN) is a popular supervised deep learning algorithm which commonly used to deal with image due to the superior of feature extraction on the convolution layer. However, there are some of the researchers got an outstanding result on the Natural Language Processing (NLP) task using CNN. Normally, a simple CNN structure will have three main layers. First is the Convolution layer that will work on extracting feature the image using the filter in order to produce feature maps. Secondly is the Pooling layer that will concatenate the value of the feature maps so that the feature dimension can be reduce without affecting the performance of the algorithms. To deal with different situation, the pooling layer is designed in different types such as max pooling, average pooling layer and sum pooling. Max pooling means gets the greatest value from the feature maps, average pooling means get the average value of the feature maps whereas sum pooling means get the total of the feature maps. Finally, is the fully connected layer. After flatten the size, it will be fed into fully connected layer to output the classification result. There is a little different while using in NLP task, the matrix of word embedding can be viewed as image matrix. Figure 2.5 shows how CNN is used in image task. The details of CNN using in NLP will be explained in chapter 3.

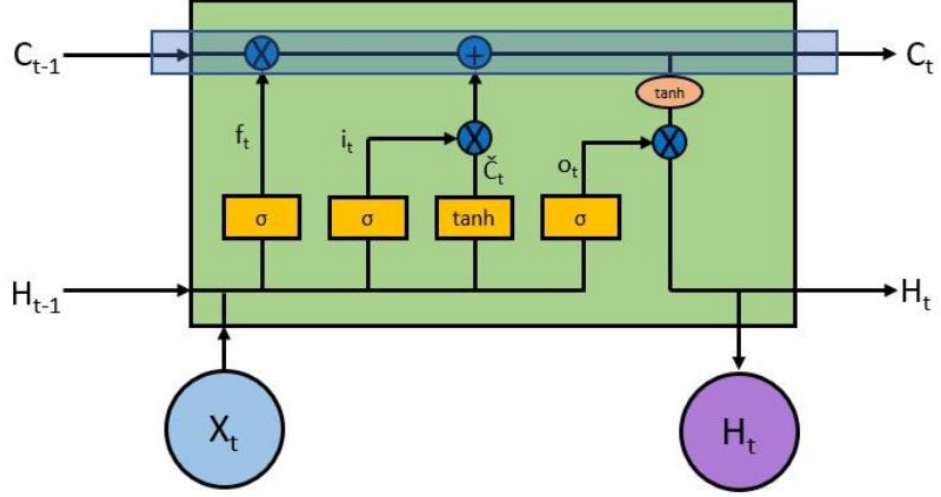


**Figure 2. 5: CNN used in image task**

### 2.4.8 LSTM

LSTM is the abbreviation of Long Short-Term Memory network. IT is one type of general Recurrent Neural Network that can solve the long-term dependency

problem in RNN. LSTM deals well with sequence modelling. The special design of LSTM, gates can increase the ability of the cell to remove or add in information. The structure of the gate is to decide how the information pass through the cell. Figure 2.6 shows the four layers of the LSTM block. It takes the current input (X) as well as the previous input (A) to produce output (H) and current state (A).



**Figure 2. 6: structure of LSTM in RNN**

There are three gates to introduce in the layer. First, Forget Gate in charge of deciding which information is unmeaningful and can be remove in the state. (2.8) is the equation.  $\sigma$  refers to the logistic sigmoid function. It contains the  $[0,1]$  values that can control the information in current state should be forgot or kept.  $b$  means the bias and  $w$  is the weight in the current state.  $h_{t-1}$  is the hidden state in each time step.

$$f_t = \sigma(w_f * [h_{t-1}, x_t] + b_f) \quad (2.8)$$

Input Gate which will make the decision to store which value and a  $\tanh$  layer will calculate the candidate values that using for updating the cell state. Sigmoid function in this state is to decide which value to store. Both Input Gate and  $\tanh$  layers are used to create the update of the state. (2.9) and (2.10) are the equations.

$$i_t = \sigma(w_i * [h_{t-1}, x_t] + b_i) \quad (2.9)$$



$$\tilde{C}_t = \tanh(w_c * [h_{t-1}, x_t] + b_c) \quad (2.10)$$

Output Gate will decide the output based on the filter version in the cell state. It will go through the sigmoid layer to decide which parts to output and tanh layer will push the value (-1 ~ 1) and multiply with the previous. Then, it will get the final output. The mathematical equation looks like (2.11) and (2.12).

$$o_t = \sigma(w_o * [h_{t-1}, x_t] + b_o) \quad (2.11)$$

$$h_t = o_t * \tanh(c_t) \quad (2.12)$$

#### 2.4.9 BERT

BERT is the abbreviation of Bidirectional Encoder Representation from Transformer. It is an open-source technique developed by Google. It is an NLP (Natural Language Processing) pre-training approach that can deal with a huge amount of text data. BERT works with Transformer to learn the relationship of the words in the text. Transformer has encoder and decoder mechanisms, but Bert only uses the encoder mechanism which is used to read the input text. Figure 2.7 shows the Bert input representation. The total of the token embeddings, segmentation embeddings, and position embeddings will be the input embeddings.

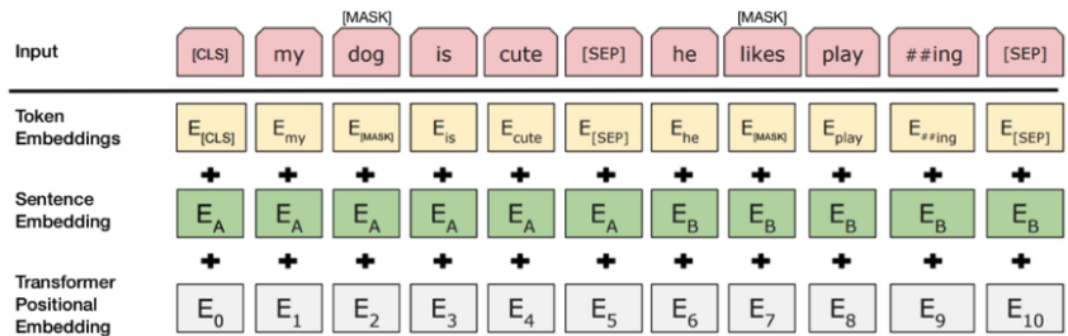


Figure 2.7: Bert input representation

## CHAPTER 3

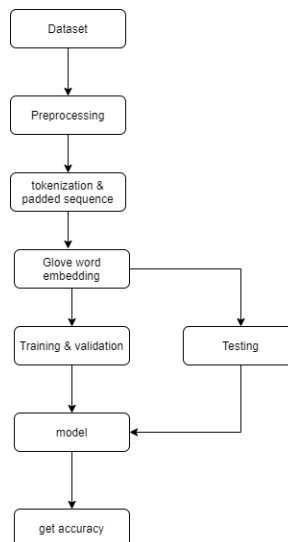
### PROPOSED MODEL

#### 3.1 Overview

In this chapter, the process and the structure of the proposed model will be presented. Section 3.2 will give an overview of the whole process. Section 3.3 will focus on the text pre-processing phase. Section 3.4 will explain how the tokenize and pad sequence the text. Section 3.5 will give an explanation of word embedding. Section 3.5 will describe the proposed solution architectures which includes CNN model and CLSTM model.

#### 3.2 Process of the model

In this section, the process of the proposed model will be briefly described. First of all, the dataset will be imported into the program to pre-process the raw text. Then, tokenization and pad sequence will be applied to the text. In this model, the GloVe word embedding is going to be used. Before feeding the data into the model, it will split into training, validation, and test data. After training the model by using training and validation data, the test data is used to predict the accuracy of the model. Figure 3.1 is the process of the proposed model.



**Figure 3. 1: Process of the proposed model**

### **3.3 Pre-processing**

To ensure that the classifiers can learn useful features from the text, the pre-processing part is needed to clean the noise of the text. First, the unnecessary column will be dropped. For example, the columns like ID or date will be dropped because it doesn't affect much for the training. Then, the data with "null" contexts will also be removed. After that, all the text will be converted to lowercase and the punctuation will also be removed to make the text clean. Lastly, stop words will be removed to ensure the model will only learn the useful feature.

#### **3.3.1 Stop words removal**

Stop words are meaningless words and will not affect much in the text, such as is, of, on, or, that, the, etc. These words normally are used to connect the sentence or make sentence structure to be fluent. However, these will be the noise of the feature in text classification, so they need to be removed to make sure the models only learn the important features.

### **3.4 Tokenization and sequence padding**

. In order to let the machine to understand the text, it is necessary to construct text into a structured form of numerical data. Tokenization is an important way for text classification. Tokenization means to break the string of text into smaller units called tokens like words, characters, or sub words so that the machine can understand and deal with it. In this project, the text will be tokenized into words by using a package called Keras tokenizer. After that, each of the tokens can be used to create the word index, in other words, it is a vocabulary corpus. Every single word has each unique integer ID starting from 1 to 20000 because the maximum number of the word has been set to 20000 like what has shown in figure 3.2(left). To make sure all the tokenized data have an equal size sequence, padding is needed. The maximum sequence length of the tokenized sentences has been set to 1000. Hence, if some of the articles didn't reach the length, the pad sequence function will add 0 value until it reaches the length. For example, figure 3.2(right) shows the tokenized vector representation from one particular article of datasets. After this step, the shape of the

data will become (amount of data, 1000). Then, the tokenized data will be split into 80% for training and 10% for validation and 10% for testing.

<b>word_index</b>	158,	4849,	28,	3643,	342,	115,	182,	1914,	182,
	410,	952,	2559,	46,	4459,	4328,	15,	287,	2559,
	202,	921,	11,	2386,	669,	473,	35219,	445,	273,
	50,	2713,	1895,	253,	143,	175,	2,	6144,	1,
	8,	762,	100,	607,	61,	152,	9,	1591,	2242,
	20,	190,	651,	143,	493,	1591,	47,	1,	39,
	39,	762,	100,	607,	1256,	362,	2221,	75,	1206,
	2019,	4,	592,	384,	45,	20,	30,	336,	100,
	1039,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	0,	0,	0,	0,	0,	0,	0,

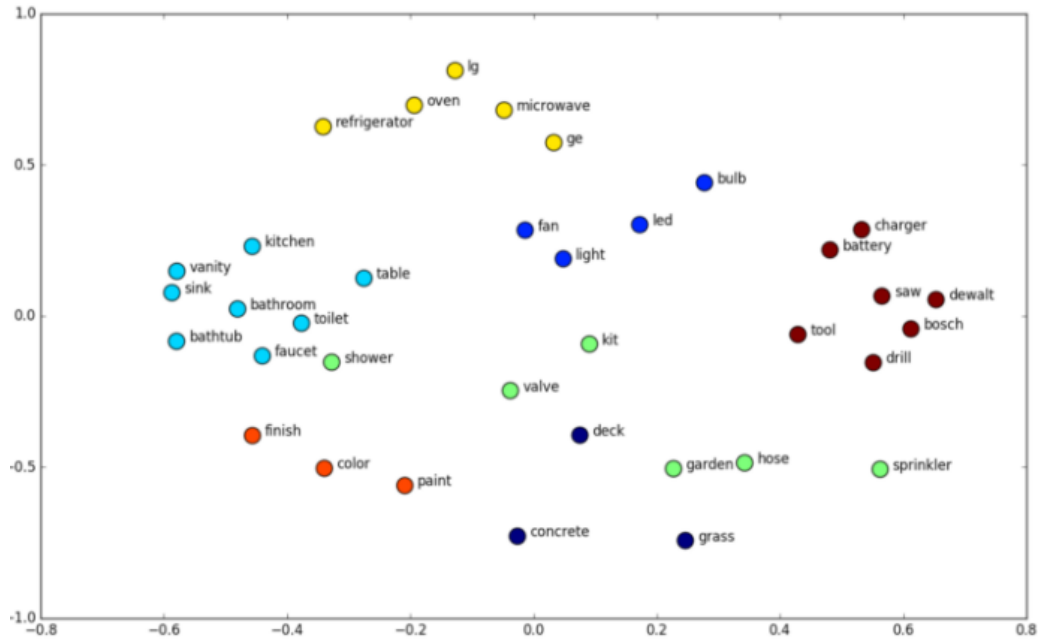
Word index

Tokenized data

**Figure 3. 2: example of tokenized sentence and word index**

### 3.5 Word Embedding

Word embedding is using the idea which maps each word in the coordinate. It means that every word has its own corresponding numeric vector. The distance among the related words will get closer instead the not related words will be far away from each other. Figure 3.3 shows the word embeddings in coordinate and it is taken online. The word "bathroom" is closed with "toilet" because they have related relationships with each other. However, the "battery" is far away from these two words because they have less related relationships. To train the word embeddings, it needs a larger dataset, but the datasets used in this project are considered as the smaller datasets, so the pre-trained weight of GloVe(Global Vectors for Word Representation) word embedding is used. This project will use the 100-dimension GloVe as the pre-trained weight.



**Figure 3. 3: Example of word embedding in geometric space**

Therefore, to use the pre-trained word vector from GloVe, an index of words is created to map into the tokenized matrix mentioned in the previous section. Then, the new word embeddings will be created with the pre-trained weight of GloVe instead of random weight. There are 400000 of word vectors contained in the Glove 6B vector 100d file. Each of the word vectors refers to a word(token), so it means that there are 40000 of the unique words inside the file. After this step, the word embeddings with the size of (1000,100) is created.

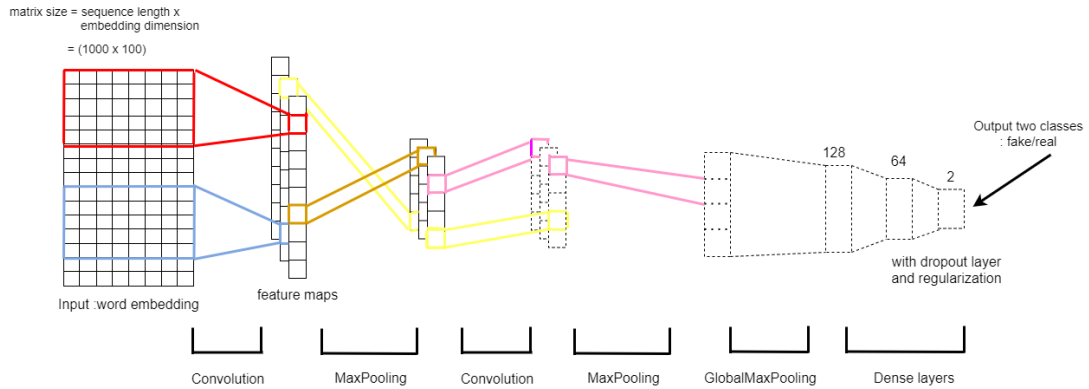
### 3.6 Model architectures

There are two section in this subtopic including CNN model and CLSTM model. Each of the section will explain and present how the architectures work in solving the fake news detection task. Both of the section will contain the figures of the architecture.

#### 3.6.1 CNN model

The concept of CNN model is trying to extract the local feature from the datasets by using the convolutional layer. This model contains embedding layer, two

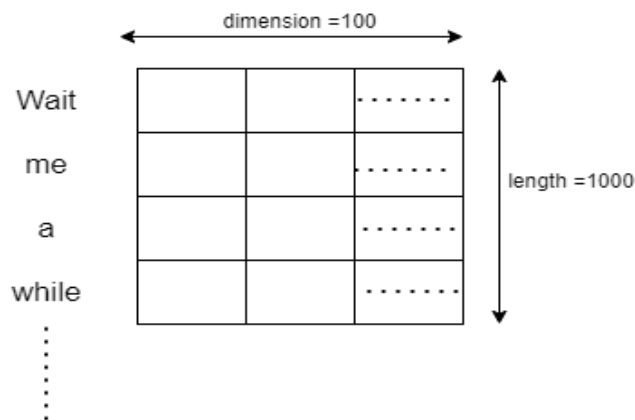
convolutional and max pooling layers, global max pooling layer and a series of dense layer. The below section will explain how these layer works in the model.



**Figure 3. 4: CNN architecture**

### Embedding layer

The embedding layer transforms words into their corresponding word embeddings matrix. The vector of word index is fed as the input into the embedding layer. Given the vector of word index with length = 1000 where every index represents a word, the weights of each word are taken from the GloVe pretrained model. The output of this step is the word embeddings matrix with the size 1000×100 where each row represents a word and the columns encode the weights. Figure 3.5 shows sample word embeddings matrix with dimension = 100 and sequence length = 1000.



**Figure 3. 5: example of word embedding**

### Convolutional layer

The core building block of CNN is convolutional layers. The convolutional layers aim to extract high level meaningful features from the input. A convolution is a linear operation that multiplies the input with a set of weights, known as a filter or a kernel. The filter is applied systematically to each filter-sized patch of the input data. The output from this operation is a feature map. Subsequently, each value in the feature map is passed through a nonlinearity activation function. The convolutional layer in this model involves a set of 128 filters with the size of  $5 \times 5$ .

In this model, a Rectified Linear Unit (ReLU) activation function is used. The main benefit of using the ReLU function is that only the neurons with value  $\geq 0$  are activated, therefore it is more computationally efficient compared to other activation functions.

### Max Pooling layer

The main purpose of max pooling layer is to reduce the dimension of the input for computational efficiency in the subsequent layers. The max pooling is a convolution process where the filter extracts the maximum value in each patch of the feature map it convolves. This model leverages the max pooling filter of size  $5 \times 5$ . The output of this layer is the set of feature maps that are reduced to 20% of the original size.

### Global Max Pooling layer

After two times of convolution and Max Pooling layers, this layer is responsible for converting the 3D tensor into 2D so that it can be fit into the Dense layer.

### Dense layers

The dense layers are also known as fully-connected layers. These layers connect all the inputs from one layer to every activation unit of the next layer. The last few dense layers in the model compile the data extracted by previous layers to form the final output. The output layer applies the “sigmoid” activation function due to the binary labelling of the data, which is fake or real news.

### Other parameters

Two regularization techniques, namely L2 regularization and dropout are also applied in the dense layers to prevent overfitting. The overfitting problem is normally caused by the over complicated network model. It causes the model to perform very well during training but performs poorly when given unseen data during testing.

The L2 regularization makes the current weights smaller in every update to reduce the impact of the hidden neurons. By doing so, the hidden neurons become negligible thus reducing the overall complexity of the network. In addition to L2 regularization, the dropout regularization is also applied. It uses a dropout with probability 0.3, where 30% of the neuron will be dropped at every update.

This model also optimizes the gradient descent process by leveraging the Adam optimizer. Adam optimizer adaptively tunes the learning rate for each weight in the network by considering the momentum. The advantage of the momentum is that it uses the moving average of the gradients hence avoid stuck in the local minima. The Adam optimizer accelerates and smooths the process of gradient descent.

As part of the optimization algorithm, a loss function is required to estimate the loss in every training epoch. In the fake news detection task, the binary cross entropy is chosen as the loss function. The binary cross entropy is calculated as below:

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \quad (3.1)$$

where  $y$  is the label (1 for fake news and 0 for real news) and  $p(y)$  is the predicted probability of the text being fake news for all  $N$  samples.

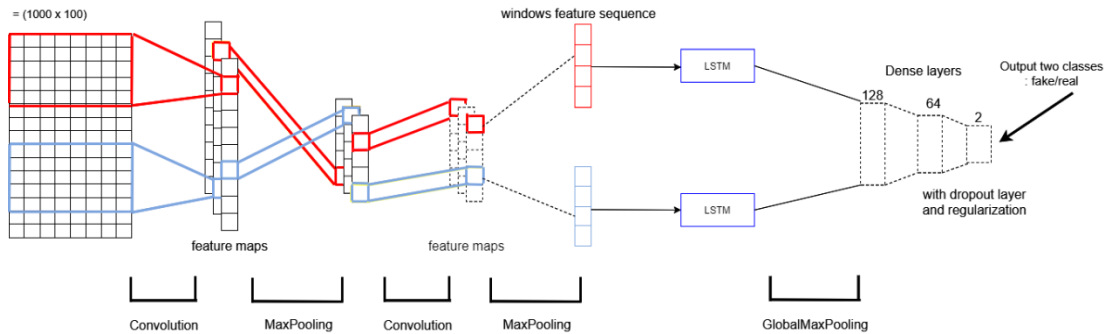


Layer (type)	Output Shape
embedding_1 (Embedding)	(None, 1000, 100)
dropout (Dropout)	(None, 1000, 100)
conv1d (Conv1D)	(None, 1000, 128)
max_pooling1d (MaxPooling1D)	(None, 200, 128)
conv1d_1 (Conv1D)	(None, 200, 128)
max_pooling1d_1 (MaxPooling1D)	(None, 40, 128)
global_max_pooling1d (GlobalMaxPooling1D)	(None, 128)
dense (Dense)	(None, 128)
dropout_1 (Dropout)	(None, 128)
dense_1 (Dense)	(None, 64)
dropout_2 (Dropout)	(None, 64)
dense_2 (Dense)	(None, 2)

**Figure 3. 6: CNN model structure**

### 3.6.2 CLSTM model

This model is using the concept of CNN which is extract the local feature representation of the text and produce the window feature sequence layer as the input of LSTM to learn the long-term dependencies according to sequence. The combination of CNN and LSTM has also brought a good performance of result. Figure 3.7 has shown the architecture. Below section will explain how this model works.



**Figure 3. 7: CLSTM architecture**

### Embedding layer

Similar as CNN model, the input of the convolutional layer in this model will be the embedding layer which used to the pretrained weight of word embedding (Glove). The size of the word embedding will also same as previous model which is 1000 of text sequence length and 100 dimension of word vector. The “trainable” variable has been set to false to let the model use the pretrained weight instead of using the random weight of word vector. The example picture can refer to figure 3.5 in the previous subtopic.

### Convolutional layer

The first layer of this model will be convolutional layer. This layer will work for producing the high-level sequence feature map as the output. Then, the value of the sequence feature map will pass through the nonlinearity activation function which is Rectified Linear Unit(ReLU) because it is more computationally efficient than others. This layer has been set to 128 filters with size of 5 x 5. This model contains two convolutional layer following by max pooling layer.

### Max pooling layer

The purpose of applying this layer is to get the maximum value of the feature map in order to reduce the dimension input for avoiding the computational burden in the following layer. This model used filter size of 5x5 in max pooling layer which means the output size after this layer will be reduced 20% and passed to next layer.

### Window feature sequence layer

The input of LSTM will be the window feature sequence layer which is produced after the feature maps. From figure 3.6, it is obviously seen that the blocks of same colour from the feature maps will produce a same colour window feature sequence in the next state.

### LSTM layer

This layer will take window feature sequence as input to proceed the next stage. The purpose of applying LSTM is to propagate the historical information via the chain of neural network and at the same time it can store the information of previous input. It is because LSTM has three gates to allow it to do so. The three gates are forget gate( $f_t$ ), input gate( $i_t$ ) and output gates( $o_t$ ). When data comes to LSTM unit, the forget gates will decide which information is unnecessary and forgettable. Input gates will decide which information is important that needs to be memorized. Whereas, output gates will decide what value in the memory cell should be the output. The equations below show the variants which will be used in this model.

$$f_t = \sigma(w_f * (V_f h_{t-1}) + b_f) \quad (3.2)$$

$$i_t = \sigma(w_i * (V_i h_{t-1}) + b_i) \quad (3.3)$$

$$o_t = \sigma(w_o * (V_o h_{t-1}) + b_o) \quad (3.4)$$

$$c_t = \tanh(w_c * (V_c h_{t-1}) + b_c) \quad (3.5)$$

W and V refers to the weight in each element. Where h is the hidden state that is related to the time step t-1. c is the cell block that can control the update value and b refers to the bias in current state. In this model, the LSTM unit will be set to 80.

### Global max pooling layer

After the LSTM layer, the data needs to be reshaped into a 2D tensor to fit into the dense layer of the following process.

### Dense layer

Same as previous model, this model contains three hidden layers with 0.02 of L2 regularizer and two dropout layers with probability of 0.3 are added between the dense layers. The top two hidden layers are using “relu” activation function whereas

the last hidden layer is using “sigmoid” activation function because there is only two label which is fake or real to classify the text.

#### Other parameters

This model is using adam optimizer. The default learning rate of this optimizer is 0.01. To classify the binary label data, the binary cross entropy is applied as the loss function. There will be 10 epochs to train data and 128 of batch size in each epoch.

Layer (type)	Output Shape
embedding_1 (Embedding)	(None, 1000, 100)
dropout (Dropout)	(None, 1000, 100)
conv1d (Conv1D)	(None, 1000, 128)
max_pooling1d (MaxPooling1D)	(None, 200, 128)
dropout_1 (Dropout)	(None, 200, 128)
conv1d_1 (Conv1D)	(None, 200, 128)
max_pooling1d_1 (MaxPooling1D)	(None, 40, 128)
lstm (LSTM)	(None, 40, 80)
global_max_pooling1d (GlobalMaxPooling1D)	(None, 80)
dense (Dense)	(None, 128)
dropout_2 (Dropout)	(None, 128)
dense_1 (Dense)	(None, 64)
dropout_3 (Dropout)	(None, 64)
dense_2 (Dense)	(None, 2)

**Figure 3. 8: CLSTM model structure**

## **CHAPTER 4**

### **Experiment**

#### **4.1 Overview**

This chapter contains six subtopic which related to the experiment details. Section 4.1 will give a brief overview of this chapter. Section 4.2 will introduce the information of the four datasets which is used in the experiment. Text analysis will be present in section 4.3 in order to understand the structure of the fake news and real news article. Section 4.4 is describing the environment of the experiment and how the experiment will be conducted. Section 4.5 presents the evaluation methods which will be used to evaluate the classification models. Lastly, section 4.6 will show the result of the models and make a discussion regarding to the 2 proposed models.

#### **4.2 Datasets**

In this project, there are three different datasets being used to test the performance of the methods. These datasets are available online and used by most of the researchers. In this section, the size, structure, and origin of datasets will be described.

##### **4.2.1 D1 (Fake or real news dataset)**

This dataset was collected by George McIntire. He gathered the data “fake news” from Kaggle that released about 13500 articles on the 2016 election cycle. The total number of “Real News” articles are about 5279 articles. These articles came from different media organizations, including the New York Times, WSJ, Bloomberg, NPR, and The Guardian that published in 2015 and 2016. Finally, this dataset was well balanced with 3171 “Real news” articles and 3164 “Fake news” articles. Hence, the total number of datasets 1 is 6335 news articles. It used “FAKE” and “REAL” to label the articles.

#### **4.2.2 D2 (ISOT dataset)**

This dataset(Ahmed et al., 2017) contains 21417 of "true news" articles in the True.csv file and 23481 of "fake news" articles Fake.csv file. Both of them have 4 columns including title, text, subject, and date. The true news is collected from Reuter.com whereas fake news articles are collected from the unreliable website that was flagged by Politifact and Wikipedia. PolitiFact is a fact-checking organization in the USA. This dataset has been categorized into different topics. True.csv has two topics, there are 'political' and 'World News', whereas Fake.csv has six topics, there are 'News' 'politics' 'Government News' 'left-news' 'US News' 'Middle-east'. This dataset has been cleaned, yet there are still some punctuation problems and mistakes remain.

#### **4.2.3 D3 (Kaggle-Getting real about fake news dataset)**

This dataset is collected by Yang et al., (2018). It contains 53 columns including title, text, image, type, author and etc. In this project, there is only text and type will be taken for the experiment. It contains 20015 articles, 11941 for fake news, and 8074 for real news. Fake news is compiled by Megan Risdal from over 240 unreliable websites. The real news is crawled from the reliable news websites like New York News and Washington Post, etc.

#### **4.2.4 D4 (Combined news dataset)**

This dataset is download from “towards data science” website and it is combined by Amol Mavuduru. This dataset is the combination of multiple dataset which are taken from Kaggle.com. It contains 74012 articles; 36969 articles belong to true news and 37043 belong to fake news. After combining the datasets into one, the author has arranged the datasets into three column which are title, text and label. It uses 0 and 1 to label fake and true news.

### 4.3 Text Analysis

This subtopic is going to discuss the difference between fake news and real news. Some simple and basic text analyses also have been done in this project, such as the average of title length and the average of text length.

**Table 4. 1: Comparison of title and text length in different dataset**

		average number of words in title	average number of words in text	D1
0	real news	9.861243	873.257647	
1	fake news	11.133059	679.129267	
		average number of words in title	average number of words in text	D2
0	real news	9.954475	385.640099	
1	fake news	14.732805	423.197905	
		average number of words in title	average number of words in text	D3
0	real news	9.986995	831.419247	
1	fake news	10.530525	648.897915	
		average number of words in title	average number of words in text	D4
0	real news	11.371714	313.633049	
1	fake news	13.354244	278.699970	

Table 4.1 has shown the analysis of the datasets. The above data shows that all of the datasets have more words in fake news titles and fewer words in real news titles. Besides, most of the words in real news text are more than fake news text. Hence, the data shows that fake news tries to use the long description of the title to attract the reader so that the average number of words in the title is longer than real news.

### 4.4 Experiment Setup

In this project, Python will be used as the platform to do the experiment. The experiment is conducted on the machine which equipped ASUS Intel Core i5-8250U 3.4Ghz with 4GB ram. The environment of python will be used in Google Colab which support version 3.7 python. The traditional methods will run on CPU whereas the deep learning methods will be run GPU which provided by Colab so that it can save a lot of training time.

Before feeding into the classifier, the datasets need some cleaning and pre-processing steps. It includes removing punctuation, stop word, numbers, and also all the words have been lowercased. Besides, the words also have been lemmatized, it means to convert the words to the root form, for example, converting “running”, “runs” and “ran” to “run”. To do the text pre-processing, the NLTK package will be used.

text	text
if there s one thing most progressives both ...	one thing progressives bernie hillary supporte...
trey gowdy asked a pointed question of former ...	trey gowdy asked pointed question former cia d...
washington (reuters) - republican senate major...	washington republican senate majority leader m...
geneva (reuters) - u.n. special envoy for syri...	geneva un special envoy syria staffan de mistu...
author ed klein told pete hegseth on fox and f...	author ed klein told pete hegseth fox friends ...
Before	After

**Figure 4. 1: Before and after of text pre-processing**

The traditional machine learning methods will use TF-IDF as the feature extraction to get the important feature and reduce the noise of the text to avoid the high computational burden. Then, the data will be fed into the classifier. This experiment will import the package of the TF-IDF vectorizer from sci-kit learn. All the traditional machine learning methods including SVM, Decision Tree, Naive Bayes, Logistic Regression, Random Forest, and Passive-Aggressive will be using the default parameter value from the sci-kit learn package.

able	according	across	act	action	actually	added
0.0	0.0	0.07269	0.0	0.000000	0.069503	0.0
0.0	0.0	0.00000	0.0	0.000000	0.000000	0.0
0.0	0.0	0.00000	0.0	0.136104	0.000000	0.0
0.0	0.0	0.00000	0.0	0.000000	0.000000	0.0

**Figure 4. 2: Part of the TF-IDF matrix**

Below subsection will give a brief explanation of the traditional machine learning methods which used in the experiment.



#### **4.4.1 SVM**

In this model, the kernel function has been set to linear and the random state value is set to 1. The class\_weight value is set to none as the datasets used in experiment is balance separated into fake and real. The setting C is using the default value which is 1. Reducing the value of C can make the model more regularization. The other parameter will use the default value from the package.

#### **4.4.2 Decision Tree**

In this classifier, the criterion function has been set to "entropy" and the splitter is using "best" strategy. Value of 42 random state is used in this classifier. The maximum depth of the tree is set to 20 whereas the class weight is set to none because most of the datasets are balance so it does not need any adjustment. The other parameters will use the default value from the sklearn.tree package.

#### **4.4.3 Naïve Bayes**

In the experiment, the multinomial Naive Bayes is chosen to deal with the data because this model is expert in handling the continuous data like text so it is suitable to use in fake news detection topic. The parameters includes alpha, fit\_prior and class\_prior are using the default value which is taken from sklearn.naive\_bayes package.

#### **4.4.4 Logistic Regression**

This model will use "auto" value in the multi\_class function because the data used in the experiment is binary labelling. The rest of the parameter also used the default value, such as, C, penalty, verbose, warm\_start and etc.

#### **4.4.5 Random Forest**

The n\_estimators of this model is set to 200. It means that it will build 200 of tree for calculation. The maximum of the depth is using value of 70.

#### 4.4.6 Passive Aggressive

In this model, the maximum iteration has been set to 50. It means that the maximum amount of the training data which fit into the model in every epoch is 50. The value of early stopping has been set to true so that it can avoid the overfitting problem because the model will stop training while the validation value is not increasing.

For the deep neural network, pre-trained word embedding, Glove with 100 dimensions will be used in this experiment. It includes CNN, LSTM, Bi-directional LSTM, CLSTM and Bert.

#### 4.4.7 CNN

In CNN, the first layer will be the input layer with the maximum sequence length 1000. Then, the following layer is the embedding layer using the GloVe pre-trained weight. After that, there will be a one-dimensional convolutional layer that contains 128 filters with a kernel size of 5 and using the activation function of relu. Then, a Global Max Pooling layer will be used to reshape the tensor to 2D. The Dense layer is defined to produce 128 sizes of the output. The final output layer will use the sigmoid activation function. The loss function is defined using "binary\_crossentropy" because the datasets are labelled in two types of labels. The "rmsprop" will be used as an optimizer. The datasets will be trained over 10 epochs in a batch size of 128.

Layer (type)	Output Shape
embedding_1 (Embedding)	(None, 1000, 100)
conv1d (Conv1D)	(None, 1000, 128)
global_max_pooling1d (Global	(None, 128)
dense (Dense)	(None, 128)
dense_1 (Dense)	(None, 2)

**Figure 4. 3: CNN Architecture**

#### 4.4.8 LSTM/bidirectional LSTM

For LSTM and bidirectional LSTM, they are the same as CNN the input layer and embedding layer which will contain maximum sequence length and using GloVe pre-trained weight. Then, a dropout layer with a probability of 0.3 will be applied. After that, LSTM or bidirectional LSTM layer with 100 neurons will be applied. A sigmoid activation function will be used in the last Dense layer. The "adam" optimizer will be used and the loss function also defined as "binary\_crossentropy".

Layer (type)	Output Shape
embeddings (Embedding)	(None, 1000, 100)
dropout (Dropout)	(None, 1000, 100)
lstm (LSTM)	(None, 100)
dense (Dense)	(None, 2)

**Figure 4. 4: LSTM and bidirectional LSTM Architecture**

#### 4.4.9 BERT

For the BERT, it has its own built-in tokenizer, Berttokenizer, so that it does not need to do pre-processing work manually. BERT is already a pre-trained model, so only some parameter tuning needs to be done. To use BERT, there are some necessary packages required to download, such as Pymagnitude, PyTorch, and transformers, etc. In this project, the "bert-base-uncased" model will be used. The learning rate has been set to 0.001 and the optimizer is "BertAdam". The batch size has been set to 32 and the early stopping also has been applied to avoid overfitting.

#### 4.4.10 CLSTM

This model is the combination of CNN model and LSTM model. First, it will take the word embedding as input into the first layer of this model which is Convolutional layer. In this layer, it is using 128 filters with the filter size of 3. It is using the "relu" activation function in the Convolutional layer. Then, following by MaxPooling layer with the pool size of 2. LSTM layer will be the next layer with 100

output dimension and following by global max pooling layer to reshape the size of the data output. After that, it will be fit into the Dense layer. In Dense layer, a sigmoid function will be applied. Finally, the “adam” optimizer and the “binary\_crossentropy” loss function will be applied.

Layer (type)	Output Shape
embedding_5 (Embedding)	(None, 1000, 100)
dropout_4 (Dropout)	(None, 1000, 100)
conv1d_4 (Conv1D)	(None, 1000, 128)
max_pooling1d_4 (MaxPooling1D)	(None, 500, 128)
lstm_3 (LSTM)	(None, 500, 100)
global_max_pooling1d (GlobalMaxPooling1D)	(None, 100)
dense (Dense)	(None, 2)

**Figure 4. 5: CLSTM Architecture**

## 4.5 Evaluation Method

To check the performance of the method, the term like accuracy, and the macro average of the precision score, recall score, and F1 score will be used as the evaluation method for the experiment. The confusion matrix and classification report had been coded after the data fit into the classifier so that the performance of the methods can be shown.

### 4.5.1 Confusion Matrix

The confusion matrix can show the amount of data that was predicted correctly or wrongly during the testing phase. The table below shows the structure of the confusion matrix in the case of fake news detection.

TP (True Positive): The model predicts fake news and the actual is fake news

FP (False Positive): The model predicts fake news and the actual is real news.

FN (False Negative): The model predicts real news and the actual is fake news.

TN (True Negative): The model predicts real news and the actual is real news.

		Predicted	
		Fake	Real
Actual	Fake	TP	FN
	Real	FP	TN

#### 4.5.2 Accuracy, Precision, Recall, and F1- score

From the accuracy, it will show how accurate the model is while detecting the fake news. 4.1 shows the formula of accuracy. Precision calculates the percentage of fake news out of total predicted fake news. In simple words, it shows how accurately the model predicts the text is fake news and it actually is fake news. 4.2 shows the formula of precision. Recall calculating the percentage of fake news out of total actual fake news. 4.3 shows the formula of recall. F1-score is the harmonic mean of precision and recall value. Hence, the value of F1-score will be affected by both of them, if precision and recall value are low, the F1-score will also be low. 4.4 shows the formula of F1-score.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

$$F1score = \frac{2 * precision * recall}{precision + recall} \quad (4.4)$$

#### 4.6 Result

This section will show the performance of each method used in the different datasets. The best performance of the method and the proposed method will be bolded.

Section 4.6.1 and section 4.6.2 will discuss what modifications have been done in the proposed method and explained how the modification improved and solved the problem which occurred in the previous model.

**Table 4. 2: Performance of each method on Dataset1**

Model	Accuracy	Precision	Recall	F1-score
SVM(Poddar et al., 2019)	88.13	88	88	88
RF(Gilda, 2018)	89.04	8	89	89
LR(Poddar et al., 2019)	88.05	88	88	88
DT(Poddar et al., 2019)	78.33	78	78	78
PA ((Mandical et al., 2020))	82.94	84	83	83
NB(Poddar et al., 2019)	83.03	84	83	83
CNN(Khan et al., 2019)	91.31	91	91	91
<b>Modified CNN</b>	<b>92.73</b>	<b>93</b>	<b>93</b>	<b>93</b>
LSTM(Khan et al., 2019)	81.35	81	81	81
CLSTM(Khan et al., 2019)	91.47	91	91	91
<b>Modified CLSTM</b>	<b>92.22</b>	<b>92</b>	<b>92</b>	<b>92</b>
Bi-LSTM(Khan et al., 2019)	80.88	81	81	81
Bert(Jwa et al., 2019)	82.63	83	83	83

**Table 4. 3: Performance of each method on Dataset2**

Model	Accuracy	Precision	Recall	F1-score
SVM	91.89	92	92	92
RF	94.65	95	95	95
LR	91.85	92	92	92
DT	87.19	87	87	87
PA	89.11	90	89	89
NB	88.58	89	89	89
CNN	97.9	98	98	98
<b>Modified CNN</b>	<b>99.39</b>	<b>99</b>	<b>99</b>	<b>99</b>
LSTM	96.45	96	96	96
CLSTM	99.15	99	99	99
<b>Modified CLSTM</b>	<b>99.45</b>	<b>99</b>	<b>99</b>	<b>99</b>
Bi-LSTM	98.08	98	98	98
Bert	94.78	95	95	95

**Table 4. 4: Performance of each method on Dataset3**

Model	Accuracy	Precision	Recall	F1-score
SVM	88.41	88	87	88
RF	89.39	89	89	89
LR	88.41	88	87	88
DT	82.87	82	82	82
PA	86.04	87	83	85
NB	81.02	82	78	79
CNN	94.45	95	94	94
<b>Modified CNN</b>	<b>95.4</b>	<b>95</b>	<b>95</b>	<b>95</b>
LSTM	92.1	92	91	92
CLSTM	94.95	95	95	95
<b>Modified CLSTM</b>	<b>95.46</b>	<b>95</b>	<b>95</b>	<b>95</b>
Bi-LSTM	89.65	92	87	89
Bert	80.88	81	81	81



**Table 4. 5: Performance of each method on Dataset4**

Model	Accuracy	Precision	Recall	F1-score
SVM	86.79	87	87	87
RF	89.73	90	90	90
LR	86.52	86	87	87
DT	82.21	82	82	82
PA	78.73	82	79	78
NB	81.70	82	82	82
CNN	96.37	96	96	96
<b>Modified CNN</b>	<b>96.43</b>	<b>96</b>	<b>96</b>	<b>96</b>
LSTM	96.28	96	96	96
CLSTM	95.36	96	95	95
<b>Modified CLSTM</b>	<b>96.37</b>	<b>96</b>	<b>96</b>	<b>96</b>
Bi-LSTM	95.60	96	96	96
Bert	72.13	72	75	72

#### 4.6.1 Discussion 1

This subtopic will describe the enhancement which has been made in the CNN model. From the table 4.2 to table 4.5 has shown that the modified CNN model has increased the accuracy compared to the original. Hence, this means that the modifications made in the new model are effective. The below figure shows the differences of original CNN and modified CNN in term of structure.

**Table 4. 6: Differences of original CNN and modified CNN**

Original CNN		Modified CNN	
Layer (type)	Output Shape	Layer (type)	Output Shape
embedding_1 (Embedding)	(None, 1000, 100)	embedding_1 (Embedding)	(None, 1000, 100)
conv1d (Conv1D)	(None, 1000, 128)	dropout (Dropout)	(None, 1000, 100)
global_max_pooling1d (Global	(None, 128)	conv1d (Conv1D)	(None, 1000, 128)
dense (Dense)	(None, 128)	max_pooling1d (MaxPooling1D)	(None, 200, 128)
dense_1 (Dense)	(None, 2)	conv1d_1 (Conv1D)	(None, 200, 128)
		max_pooling1d_1 (MaxPooling1	(None, 40, 128)
		global_max_pooling1d (Global	(None, 128)
		dense (Dense)	(None, 128)
		dropout_1 (Dropout)	(None, 128)
		dense_1 (Dense)	(None, 64)
		dropout_2 (Dropout)	(None, 64)
		dense_2 (Dense)	(None, 2)

Firstly, the modified CNN has increased the 1D convolutional layer to two and following by max pooling layer. These additions can help the model to extract more detail features of the text and train the data deeper during the training. For the fully connected layer, the modified CNN has added more Dense layer into the model whereas the original CNN only has one dense layer. There are three dense layers in the modified CNN. Each of the layer contains different number of neurons which is 128, 64 and 2. This enhancement can reduce the loss of important feature in training. The two dropout layers with 0.3 of probability in between the dense layers is helping to mitigate the overfitting problem. It can help to reduce the complexity in the network.

The modification not only made on the structure of the model, the parameters in the model also have been changed. The loss function has changed from rmsprop optimizer to adam optimizer. It is because rmsprop optimizer cannot maintain the average of past gradient. Besides that, the L2 regularization with value of 0.01 is added into the model to prevent overfitting problem.

With all these modifications, the modified model has achieved a higher accuracy than the original CNN model. The plot loss figures also show the differences. The training loss and validation loss of the modified CNN has become stable compared

to the original CNN. It means that it has solved the overfitting problem. Therefore, the result has shown that modified CNN model is improved.

#### 4.6.2 Discussion 2

From the result show in table 4.2 to 4.6, the modified CLSTM model has achieved a higher accuracy in each dataset. It is because the additional layer and the modified parameters in the structure are working efficiently for the data text. Hence, this subtopic will discuss and make the analysis in term of the modified CLSTM model to understand how the modification works. Table 4.7 shows the differences of the original and modified CLSTM model.

**Table 4. 7: Differences of the original and modified CLSTM model**

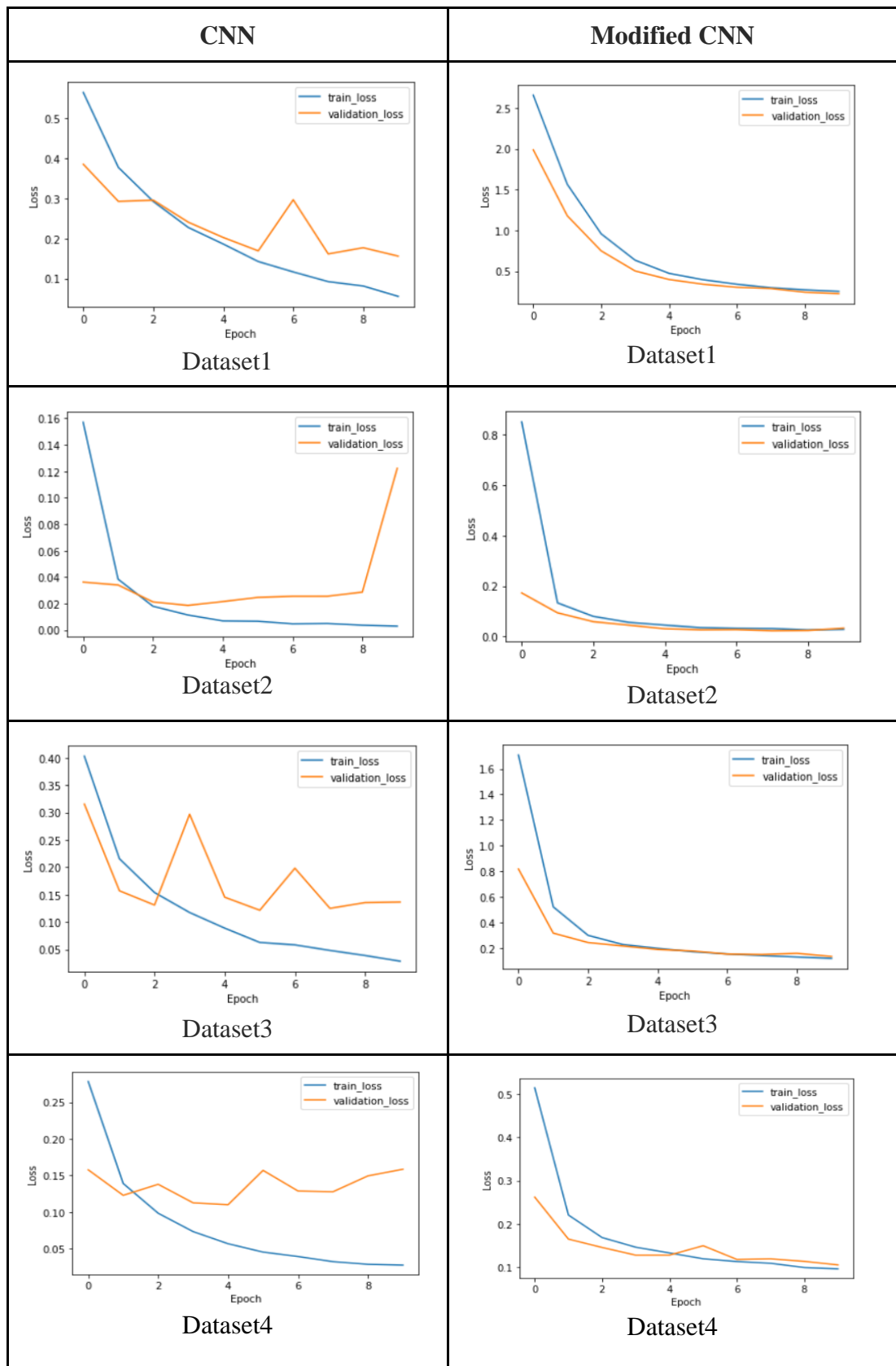
Original CLSTM		Modified CLSTM	
Layer (type)	Output Shape	Layer (type)	Output Shape
embedding_5 (Embedding)	(None, 1000, 100)	embedding_1 (Embedding)	(None, 1000, 100)
dropout_4 (Dropout)	(None, 1000, 100)	dropout (Dropout)	(None, 1000, 100)
conv1d_4 (Conv1D)	(None, 1000, 128)	conv1d (Conv1D)	(None, 1000, 128)
max_pooling1d_4 (MaxPooling1D)	(None, 500, 128)	max_pooling1d (MaxPooling1D)	(None, 200, 128)
lstm_3 (LSTM)	(None, 500, 100)	dropout_1 (Dropout)	(None, 200, 128)
global_max_pooling1d (Global)	(None, 100)	conv1d_1 (Conv1D)	(None, 200, 128)
dense (Dense)	(None, 2)	max_pooling1d_1 (MaxPooling1D)	(None, 40, 128)
		lstm (LSTM)	(None, 40, 80)
		global_max_pooling1d (Global)	(None, 80)
		dense (Dense)	(None, 128)
		dropout_2 (Dropout)	(None, 128)
		dense_1 (Dense)	(None, 64)
		dropout_3 (Dropout)	(None, 64)
		dense_2 (Dense)	(None, 2)

This model is the combination of CNN and LSTM where CNN can extract meaningful sequence feature and LSTM can learning long term dependency of the sequence feature. In CNN part, the single convolutional layer has been increased to two convolutional layer following max pooling layer. This addition helps the model to extract more useful sequence feature maps to train in the following layer.

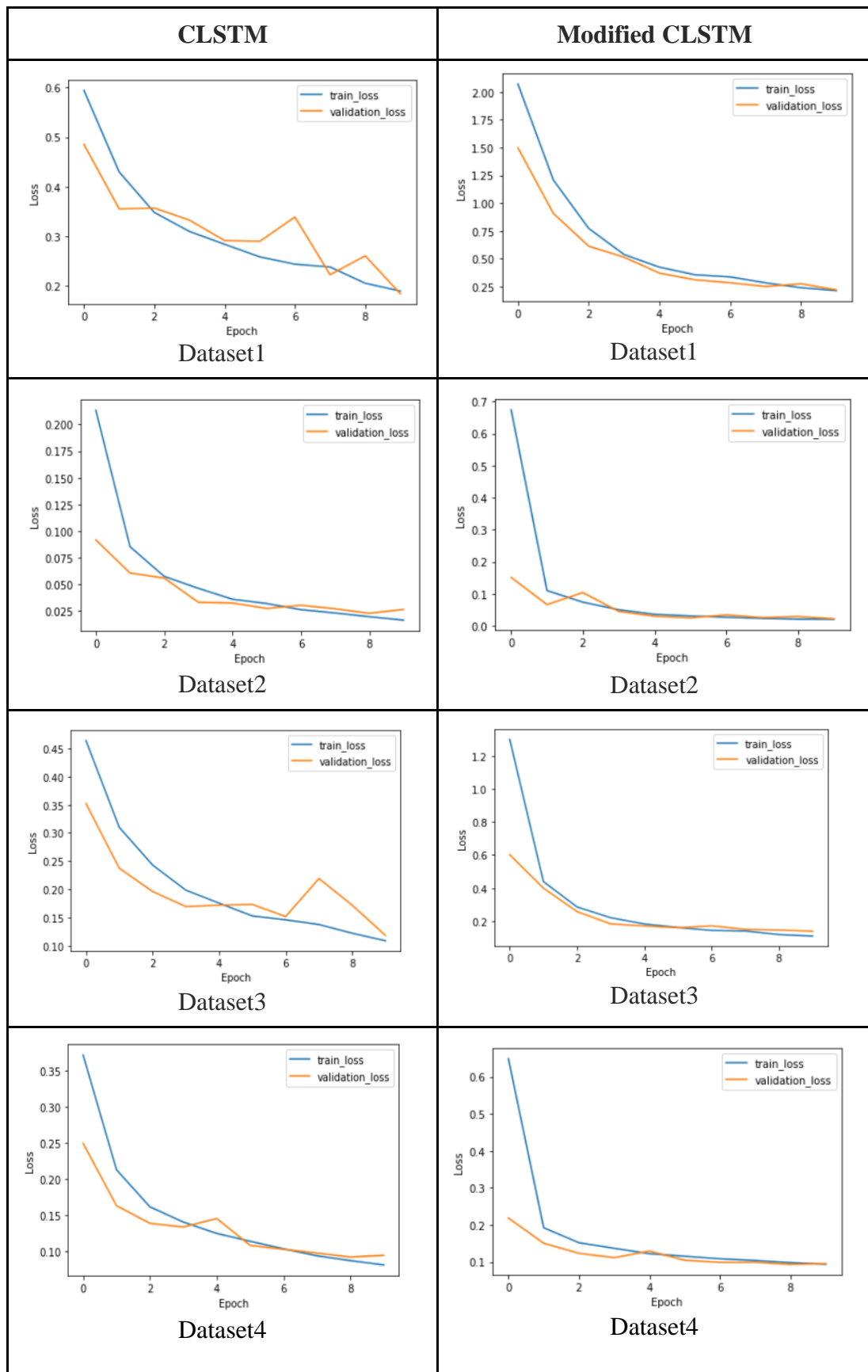
Next, the unit of LSTM has been decreased to 80. Theoretically, reducing the units of the LSTM can mitigate the complexity of the network and the computation burden in the training. Hence, the model can learn effectively during training.

Then, the fully connected layer has been added to three dense layers with 128, 64 and 2 of neurons in each hidden layer. The purpose of increment in the dense layers is to reduce the loss of the important feature. The increment of the dropout layers is to reduce the complexity in the network so that the overfitting problem can be prevent. The dropout probability of 0.3 means the network will randomly throw away 30% of the neuron in each update.

The optimizer parameter is still using the same which is adam optimizer because it can prevent the stuck in local minima. All these modifications have improved the performance of the model and reduce the overfitting problem. Although the overfitting problem in the original is not serious, the modification has brought a more stable new model. This result has been shown in the figure 4.7 with the plots that stated the loss of training and validation in every epoch. Therefore, the modification has enhanced the model effectively.



**Figure 4. 6: Plots of the loss (CNN)**



**Figure 4. 7: Plots of the loss (CLSTM)**

## **CHAPTER 5**

### **CONCLUSION**

#### **5.1 Conclusion**

In conclusion, information technology has greatly reshaped our daily life. On one hand, these advancements in information technology have made communication easier than ever. On the other hand, the misuse of information technology has also made the fake news diffuse deeper and more rapidly. Therefore, this project is intended to solve this problem. The three objectives which stated in previous chapter have been achieved and fulfilled. Firstly, the first objective is to study and analyse the existing classification methods which used to detect the fake news. To achieved it, 17 conference papers have been summarized into literature review so that the details and techniques which used by the other researchers can be taken as references. Besides, it also helps to analysis how to deal with the fake news(text).

The next objective is to enhance the performance of the existing methods. From the CNN model and CLSTM model have shown the serious overfitting problem although they have a good accuracy. In the experiment, the architecture and hyperparameters of these two models have been modified to get a better accuracy and reduce the overfitting problem. To ensure the model has been improved and stable, both of the model has been run for five times to test the accuracy from each dataset and get the average of the accuracy. Hence, these two models are the proposed models in this project. The plot loss of the two proposed model also has been presented in chapter 4 to proof that the modification in the model really improve the performance of the model.

The last objective is to evaluate the performance of the models so that people can know which method is good dealing with fake news detection. In order to achieve it, there are 11 existing methods and the 2 enhanced models have been run in four different datasets to compare the accuracy. The existing methods include 6 traditional machine learning methods and 5 deep learning methods. The result of the comparison

is shown in the tables in chapter 4. From the given accuracy in this project, modified CNN model and modified CLSTM model get a promising result after comparing with others. It is because deep neural networks can train the model deeper so that the model can captured the pattern of fake news more accurately.

## **5.2 Future work**

After completing all the objective in the project, there are some future works can be explored which may help to improve the result in this topic. For the dataset part, this project currently only used the dataset will binary labelling. In future, the multi-labelling dataset can be added in the experiment. Besides of TF-IDF vectorizer, hashing vectorizer, count vectorizer and chi-square also can be tried as feature extraction to compare whether which method is most suitable to solve this topic. Finally, there are also some other deep learning methods can be tried in the experiment.



## REFERENCES

- Agarwalla, K., Nandan, S., Nair, V. A., & Deva Hema, D. (2019). Fake news detection using machine learning and natural language processing. *International Journal of Recent Technology and Engineering*, 7(6), 844–847.
- Ahmed, H., Traore, I., & Saad, S. (2017). Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. *First International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, 10618, 169–181. <https://doi.org/10.1007/978-3-319-69155-8>
- Amine, B. M., Drif, A., & Giordano, S. (2019). Merging deep learning model for fake news detection. *2019 International Conference on Advanced Electrical Engineering, ICAEE 2019*, 5–8. <https://doi.org/10.1109/ICAEE47123.2019.9015097>
- Bahad, P., Saxena, P., & Kamal, R. (2019). Fake News Detection using Bi-directional LSTM-Recurrent Neural Network. *Procedia Computer Science*, 165(2019), 74–82. <https://doi.org/10.1016/j.procs.2020.01.072>
- Benamira, A., Devillers, B., Lesot, E., Ray, A. K., Saadi, M., & Malliaros, F. D. (2019). Semi-supervised learning and graph neural networks for fake news detection. *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2019*, 568–569. <https://doi.org/10.1145/3341161.3342958>
- Gilda, S. (2018). Evaluating machine learning algorithms for fake news detection. *IEEE Student Conference on Research and Development: Inspiring Technology for Humanity, SCORED 2017 - Proceedings*, 2018-Janua(December), 110–115. <https://doi.org/10.1109/SCORED.2017.8305411>
- Girgis, S., & Gadallah, M. (2018). ' Hhs / Hduqlqj \$ Ojrulwkp v Iru ' Hwhfwlqj ) Dnh 1Hzv Lq 2Qolqh 7H [ W. 2018 13th International Conference on Computer Engineering and Systems (ICCES), 93–97.
- Granik, M., & Mesyura, V. (2017). Fake news detection using naive Bayes classifier. *2017 IEEE 1st Ukraine Conference on Electrical and Computer Engineering, UKRCON 2017 - Proceedings*, 900–903. <https://doi.org/10.1109/UKRCON.2017.8100379>
- Hiramath, C. K., & Deshpande, G. C. (2019, July). Fake News Detection Using Deep Learning Techniques. In *2019 1st International Conference on Advances in Information Technology (ICAIT)* (pp. 411-415). IEEE.
- Jwa, H., Oh, D., Park, K., Kang, J. M., & Lim, H. (2019). exBAKE: Automatic fake news detection model based on Bidirectional Encoder Representations from Transformers (BERT). *Applied Sciences (Switzerland)*, 9(19), 1–9. <https://doi.org/10.3390/app9194062>
- Kaliyar, R. K. (2018). Fake news detection using a deep neural network. *2018 4th International Conference on Computing Communication and Automation, ICCCA 2018*, 1–7. <https://doi.org/10.1109/CCAA.2018.8777343>
- Kaur, S., Kumar, P., & Kumaraguru, P. (2020). Automating fake news detection system using multi-level voting model. *Soft Computing*, 24(12), 9049–9069. <https://doi.org/10.1007/s00500-019->

- Kesarwani, A., Chauhan, S. S., & Nair, A. R. (2020). Fake News Detection on Social Media using K-Nearest Neighbor Classifier. *Proceedings of the 2020 International Conference on Advances in Computing and Communication Engineering, ICACCE 2020*, 20–23. <https://doi.org/10.1109/ICACCE49060.2020.9154997>
- Khan, J. Y., Khondaker, M. T. I., Iqbal, A., & Afroz, S. (2019). *A Benchmark Study on Machine Learning Methods for Fake News Detection*. 1–14. <http://arxiv.org/abs/1905.04749>
- Mandical, R. R., Mamatha, N., Shivakumar, N., Monica, R., & Krishna, A. N. (2020). Identification of Fake News Using Machine Learning. *Proceedings of CONECCT 2020 - 6th IEEE International Conference on Electronics, Computing and Communication Technologies*. <https://doi.org/10.1109/CONECCT50063.2020.9198610>
- Ozbay, F. A., & Alatas, B. (2020). Fake news detection within online social media using supervised artificial intelligence algorithms. *Physica A: Statistical Mechanics and Its Applications*, 540, 123174. <https://doi.org/10.1016/j.physa.2019.123174>
- Poddar, K., Amali, G. B. D., & Umadevi, K. S. (2019). Comparison of Various Machine Learning Models for Accurate Detection of Fake News. *2019 Innovations in Power and Advanced Computing Technologies, i-PACT 2019*, 1–5. <https://doi.org/10.1109/i-PACT44901.2019.8960044>
- Thakur, A., Shinde, S., Patil, T., Gaud, B., & Babanne, V. (2020). MYTHYA: Fake News Detector, Real Time News Extractor and Classifier. *Proceedings of the 4th International Conference on Trends in Electronics and Informatics, ICOEI 2020, Icoei*, 982–987. <https://doi.org/10.1109/ICOEI48184.2020.9142971>
- Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z., & Yu, P. S. (2018). *TI-CNN: Convolutional Neural Networks for Fake News Detection*. <http://arxiv.org/abs/1806.00749>

## **APPENDICES**

## Appendix A: Meeting logs



### Faculty of Information Science and Technology (FIST) Final Year Project Meeting Log


<b>MEETING DATE: 9 December 2020</b>	<b>MEETING NO.: 1</b>
<b>PROJECT ID: T741435</b>	
<b>PROJECT TITLE: FAKE NEWS DETECTION USING MACHINE LEARNING</b>	
<b>SESSION: 2020/2021</b>	<b>SUPERVISOR: Lee Chin Poo</b>
<b>STUDENT ID &amp; Name: 1181300023 Tan Kian Long</b>	<b>CO- SUPERVISOR: Lim Kian Ming</b>

All to be filled in by student

<b>1. WORK DONE</b> Add content into chapter 2
<b>2. WORK TO BE DONE</b> find a larger dataset to train and test model
<b>3. PROBLEMS ENCOUNTERED</b> Some algorithms are difficult to understand
<b>4. COMMENTS</b> Good progress

  
**DR. LEE CHIN POO**  
SENIOR LECTURER  
Faculty of Information Science and Technology  
Multimedia University  
Jalan Ayer Keroh Lama  
75450 Melaka

  
**DR. LIM KIAN MING**  
Lecturer  
Faculty of Information Science & Technology  
Multimedia University  
Jalan Ayer Keroh Lama  
75450 Melaka, Malaysia

  
*Kian Long*

Supervisor's Signature &  
Stamp

Co-Supervisor's Signature  
& Stamp (if any)

Student's Signature

#### NOTES:

- Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
- For FYP Phase 1, total six log sheets are to be submitted (every other week\*).
- For FYP Phase 2, total six log sheets are to be submitted (every other week\*\*).
- Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

\*\* : week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



Faculty of Information Science and Technology (FIST)  
**Final Year Project Meeting Log**

<b>MEETING DATE: 23 December 2020</b>	<b>MEETING NO.:2</b>
<b>PROJECT ID: T741435</b>	
<b>PROJECT TITLE: FAKE NEWS DETECTION USING MACHINE LEARNING</b>	
<b>SESSION: 2020/2021</b>	<b>SUPERVISOR: Lee Chin Poo</b>
<b>STUDENT ID &amp; Name: 1181300023 Tan Kian Long</b>	<b>CO- SUPERVISOR: Lim Kian Ming</b>

All to be filled in by student

<b>1. WORK DONE</b> Run the new dataset through all model in experiment
<b>2. WORK TO BE DONE</b> Enhanced another model
<b>3. PROBLEMS ENCOUNTERED</b> the dataset has a lot of unnecessary columns that need to clean before fed into models
<b>4. COMMENTS</b> Good progress

  
**DR. LEE CHIN POO**  
 SENIOR LECTURER  
 Faculty of Information Science and Technology  
 Multimedia University  
 Jalan Ayer Keroh Lama  
 75450 Melaka

  
**DR. LIM KIAN MING**  
 Lecturer  
 Faculty of Information Science & Technology  
 Multimedia University  
 Jalan Ayer Keroh Lama  
 75450 Melaka, Malaysia

*Kian Long*

Supervisor's Signature &  
Stamp

Co-Supervisor's Signature  
& Stamp (if any)

Student's Signature

**NOTES:**

5. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
6. For FYP Phase 1, total six log sheets are to be submitted (every other week\*).
7. For FYP Phase 2, total six log sheets are to be submitted (every other week\*\*).
8. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

\*\* : week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



Faculty of Information Science and Technology (FIST)  
**Final Year Project Meeting Log**

<b>MEETING DATE: 6 January 2021</b>	<b>MEETING NO.:3</b>
<b>PROJECT ID: T741435</b>	
<b>PROJECT TITLE: FAKE NEWS DETECTION USING MACHINE LEARNING</b>	
<b>SESSION: 2020/2021</b>	<b>SUPERVISOR: Lee Chin Poo</b>
<b>STUDENT ID &amp; Name: 1181300023 Tan Kian Long</b>	<b>CO- SUPERVISOR: Lim Kian Ming</b>

All to be filled in by student

<b>1. WORK DONE</b> The enhanced model get a higher accuracy
<b>2. WORK TO BE DONE</b> Compile all result into report
<b>3. PROBLEMS ENCOUNTERED</b> Having some difficulty in parameter tuning
<b>4. COMMENTS</b> Good progress

  
**DR. LEE CHIN POO**  
 SENIOR LECTURER  
 Faculty of Information Science and Technology  
 Multimedia University  
 Jalan Ayer Keroh Lama  
 75450 Melaka

  
**DR. LIM KIAN MING**  
 Lecturer  
 Faculty of Information Science & Technology  
 Multimedia University  
 Jalan Ayer Keroh Lama  
 75450 Melaka, Malaysia

*Kian Long*

\_\_\_\_\_  
 Supervisor's Signature &  
 Stamp

\_\_\_\_\_  
 Co-Supervisor's Signature  
 & Stamp (if any)

\_\_\_\_\_  
 Student's Signature

**NOTES:**

9. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
10. For FYP Phase 1, total six log sheets are to be submitted (every other week\*).
11. For FYP Phase 2, total six log sheets are to be submitted (every other week\*\*).
12. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

\*\* : week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



Faculty of Information Science and Technology (FIST)  
**Final Year Project Meeting Log**

<b>MEETING DATE: 20 January 2021</b>	<b>MEETING NO.:4</b>
<b>PROJECT ID: T741435</b>	
<b>PROJECT TITLE: FAKE NEWS DETECTION USING MACHINE LEARNING</b>	
<b>SESSION: 2020/2021</b>	<b>SUPERVISOR: Lee Chin Poo</b>
<b>STUDENT ID &amp; Name: 1181300023 Tan Kian Long</b>	<b>CO- SUPERVISOR: Lim Kian Ming</b>

All to be filled in by student

<b>1. WORK DONE</b> Understand the structure of the new enhanced model and write in the report
<b>2. WORK TO BE DONE</b> Complete the chapter 4 discussion part
<b>3. PROBLEMS ENCOUNTERED</b> Structure of neural network is difficult to explain.
<b>4. COMMENTS</b> Good progress

  
**DR. LEE CHIN POO**  
 SENIOR LECTURER  
 Faculty of Information Science and Technology  
 Multimedia University  
 Jalan Ayer Keroh Lama  
 75450 Melaka

  
**DR. LIM KIAN MING**  
 Lecturer  
 Faculty of Information Science & Technology  
 Multimedia University  
 Jalan Ayer Keroh Lama  
 75450 Melaka, Malaysia

*Kian Long*

Supervisor's Signature &  
Stamp

Co-Supervisor's Signature  
& Stamp (if any)

Student's Signature

**NOTES:**

13. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
14. For FYP Phase 1, total six log sheets are to be submitted (every other week\*).
15. For FYP Phase 2, total six log sheets are to be submitted (every other week\*\*).
16. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

\*\* : week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



Faculty of Information Science and Technology (FIST)  
Final Year Project Meeting Log

<b>MEETING DATE: 27 January 2021</b>	<b>MEETING NO.:5</b>
<b>PROJECT ID: T741435</b>	
<b>PROJECT TITLE: FAKE NEWS DETECTION USING MACHINE LEARNING</b>	
<b>SESSION: 2020/2021</b>	<b>SUPERVISOR: Lee Chin Poo</b>
<b>STUDENT ID &amp; Name: 1181300023 Tan Kian Long</b>	<b>CO- SUPERVISOR: Lim Kian Ming</b>

All to be filled in by student

<b>1. WORK DONE</b> Finalize the final report
<b>2. WORK TO BE DONE</b> Extract the report content into conference paper
<b>3. PROBLEMS ENCOUNTERED</b> Some of the result get from experiment are difficult to explain.
<b>4. COMMENTS</b> Good progress

  
**DR. LEE CHIN POO**  
 SENIOR LECTURER  
 Faculty of Information Science and Technology  
 Multimedia University  
 Jalan Ayer Keroh Lama  
 75450 Melaka

  
**DR. LIM KIAN MING**  
 Lecturer  
 Faculty of Information Science & Technology  
 Multimedia University  
 Jalan Ayer Keroh Lama  
 75450 Melaka, Malaysia

*Kian Long*

Supervisor's Signature &  
Stamp

Co-Supervisor's Signature  
& Stamp (if any)

Student's Signature

**NOTES:**

17. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
18. For FYP Phase 1, total six log sheets are to be submitted (every other week\*).
19. For FYP Phase 2, total six log sheets are to be submitted (every other week\*\*).
20. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

\*\* : week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)





Faculty of Information Science and Technology (FIST)  
**Final Year Project Meeting Log**

<b>MEETING DATE: 3 February 2021</b>	<b>MEETING NO.:6</b>
<b>PROJECT ID: T741435</b>	
<b>PROJECT TITLE: FAKE NEWS DETECTION USING MACHINE LEARNING</b>	
<b>SESSION: 2020/2021</b>	<b>SUPERVISOR: Lee Chin Poo</b>
<b>STUDENT ID &amp; Name: 1181300023 Tan Kian Long</b>	<b>CO- SUPERVISOR: Lim Kian Ming</b>

All to be filled in by student

<b>1. WORK DONE</b> Complete the conference paper
<b>2. WORK TO BE DONE</b> Planning what should include in presentation
<b>3. PROBLEMS ENCOUNTERED</b> Some techniques are difficult to summarize.
<b>4. COMMENTS</b> Good progress

  
**DR. LEE CHIN POO**  
 SENIOR LECTURER  
 Faculty of Information Science and Technology  
 Multimedia University  
 Jalan Ayer Keroh Lama  
 75450 Melaka

  
**DR. LIM KIAN MING**  
 Lecturer  
 Faculty of Information Science & Technology  
 Multimedia University  
 Jalan Ayer Keroh Lama  
 75450 Melaka, Malaysia

*Kian Long*

\_\_\_\_\_  
 Supervisor's Signature &  
 Stamp

\_\_\_\_\_  
 Co-Supervisor's Signature  
 & Stamp (if any)

\_\_\_\_\_  
 Student's Signature

**NOTES:**

21. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
22. For FYP Phase 1, total six log sheets are to be submitted (every other week\*).
23. For FYP Phase 2, total six log sheets are to be submitted (every other week\*\*).
24. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

\*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

\*\* : week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)

## Appendix B: Checklist



Faculty of Information Science and Technology (FIST)

### Checklist for Final Report Submission

(To be filled in by Student)

#### STUDENT'S DETAILS

Project Code	T741435
Name	Tan Kian Long
ID No	1181300023
Title of Thesis	Fake new detection using machine learning
Supervisor Name	Lee Chin Poo

REPORT ARRANGEMENT	√	Comments (if any differences)
1. Cover of The Final Report	√	
2. Title Page of the Final Report	√	
3. Copyright page of I Final Report	√	
4. Declaration Page of Final report	√	
5. Acknowledgement	√	
6. Table of Contents	√	
7. Abstract	√	
8. List of Tables	√	
9. List of Figures	√	
10. List of Symbols	√	
11. List of Appendices	√	
12. Chapter 1: Introduction – objectives, scope	√	
13. Chapter 2: Literature Review	√	
14. Chapter 3: Proposed Model	√	
15. Chapter 4: Experiment	√	
16. Chapter 5: Conclusion	√	
17. Chapter 6: Title: -	√	
18. Chapter 7: Title: -	√	
19. Chapter 8: Title: -	√	
20. References – APA style	√	
21. Appendices	√	
22. CD/ DVD and envelope as shown in Appendix K	√	
23. Attachment: FYP Meeting Logs (all) - 1 set	√	

FORMAT OF REPORT	√	Comments
1. Page Numbering	√	
2. Font and Type Face	√	
3. Font Cover	√	
4. Tables and Figures	√	
5. Spine Format	√	
6. Comb Bind (For evaluation)	√	
7. Permanent Bind (After approval)	√	
8. Colour of the Front Cover	√	
9. Number of words > 10000 (Main content only)	√	

Checked by

Kian Long 18/2/2021

Student's Signature & Date

