Detection And Performance Evaluation of Online-Fraud Using Deep Learning Algorithms

N, Dr. Megharani Patil
Department of Computer Engineering, Thakur College of Engineering &Technology, Mumbai, India
anam.mail4u@gmail.com, megharani.patil@thakureducation.org

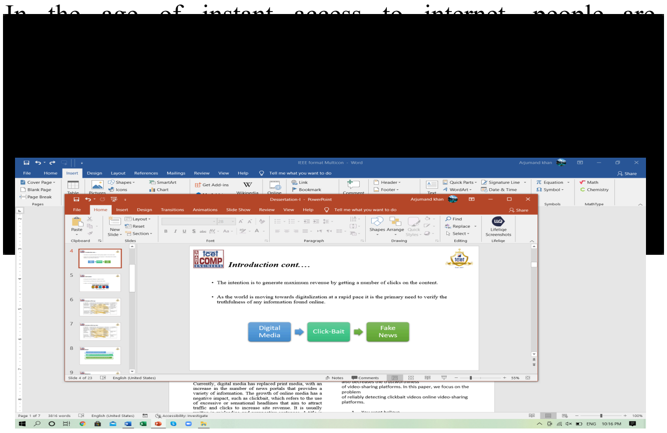
Abstract—**Online News Portals arecurrently one of the primary sources used by people, though its credibility is under serious question. Because the problem associated with this is Click-bait. Click-baiting being the growing phenomenon on internet has the potential to intentionally mislead and attract online viewership thereby earning considerable revenue for the agencies providing such false information. There is need for accurately detecting such events on online-platforms before the user becomes a victim.**
research in this field is still active in trying to enhance the accuracy of these
systems. In this paper, we will use supervised machine learning.
research in this field is still active in trying to enhance the accuracy of these
systems. In this paper, we will use supervised machine learning.
research in this field is still active in trying to enhance the accuracy of these
systems. In this paper, we will use supervised machine learning.
, research in this field is still active in trying to enhance the accuracy of these
systems. In this paper,
, research in this field is still active in trying to enhance the accuracy of these
systems. In this paper,
, research in this field is still active in trying to enhance the accuracy of these
systems. In this paper,
The solution incorporates a Novel Neural Network Approach based on FastText Word2Vec Embeddings provided by Facebook and Natural Language Processing where Headlines are specifically taken into consideration. The proposed system consists of Hybrid Bi- Directional LSTM-CNN model and MLP model. Promising Results have been achieved when tested on a dataset of 32,000 columns equally distributed as Click-bait and Non-Click-bait, in terms of Accuracy, Precision and Recall. The graphs achieved are also self-explanatory in terms of reliability of the system. A comparative analysis is also been done to show the effectiveness of our model in terms of detecting Click-bait which is heavily present on-line.

Keywords—Click-bait, FastText, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Multi-Layer Perceptron (MLP)

I. Introduction

Currently, print media is replaced by digital media, resulting in the increasing number of news portals that provides a number of information. The growth of online media led to clickbait, which is a negative impact of online journalism referring to the use of excessive or sensational headlines with only aim to attract traffic and clicks to increase site revenue. It is usually written in a language which is misleading and has provocative sentences. A title gives an initial impression and influences the user's perception and is an essential element in the news. A theory proposed by Lowenstein states that clickbait is formed by knowledge gaps created by one's curiosity in certain matters. This gap is capable of affecting one's emotions.

In the age of instant access to internet, people are



Introduction cont….

- The intention is to generate maximum revenue by getting a number of clicks on the content.
- As the world is moving towards digitalization at a rapid pace it is the primary need to verify the truthfulness of any information found online.

Figure 1: Overview of Clickbait

The time spent by peopleon online media is expected to be much more than thetime they spend on traditional TV worldwide in the year 2019.Considering, YouTubehas more than a billion users covering almost one-third of the Internet population andreaches billions of views per day. An online newstypically is made up of title,thumbnail, and the video content. Before the news the title and thumbnail (in case of video) are visible to theviewers, before they click and actually identify the content. Hence, headlines are found to be the crucial factors thatattracts the users to click and watch any video.The content is clearly different from its title or thumbnail.The content is specially generated to attract viewers to click videoand increase the viewership of the video. However, spreading of clickbait videoswastes the time of viewers also decreases the trustworthiness towards journalism. Some of the common examples of clickbait headlines are:

1. ***You won't believe…...***
2. ***These 12 tricks ……will change your life……***
3. ***Omg!!! Click to see what happens next….***

Using intentionally deceiving links, tweets, or social media posts to attract onlineviewership, are all strategies of click baiting and it has been one method of flooding misinformation on the internet.A lot of attention has come towardsClick-bait even though research in the field of clickbait detection is still in an early phase.Because of the extensive use of clickbaitin online media and news, significant fallout has startedto happen against social media platforms where any such content is found. Social Media platforms such as Facebook decided to take action against such clickbait activities, however it still continues to be flooded with such articles. To fight against this, huge number ofTwitter handles have emerged and gained number of followings, with only purpose to identify clickbait. Handles such as @SavedYouAClickand @HuffPoSpoilers are consistentlyupdating their feeds with such posts to create awarenessabout them.

The method is bit time taking because of their manual detection as users running those accounts themselves read andclassify the tweet as clickbait or not for the benefit of people. According to sources sentimental headlines create more curiosity among people and leads to clickbait.Around 69, 000 headlines from four internationalmedia houses in 2014 were analyzed based on polarity of sentimentsand found extremities in sentimentsresulted in increased popularity.

Headlines are the firstimpression and it can affect how the news articles are considered by users.

A headline strongly affects which existing knowledge is triggered in one's brain.By its way of phrasing, a headline can dominate one'smindset so that readers later recall details that coincide withwhat they were expecting, leading individuals to perceive thesame content differently according to the headline.

Another explanation isthe frequently said Loewenstein's information gap theory.In simple words, the theory states that whenever we distinguisha gap between what we already know and what is unknown to us,that gap has emotional consequences. Such information gaps lead us towards false content provided online.

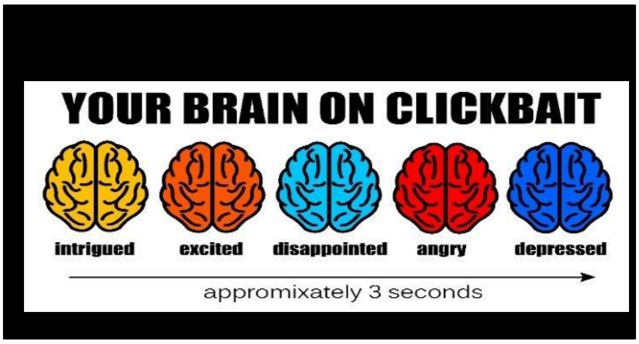


Figure 2: Effect of clickbait on mental state

II. Literature Survey

The click-bait detection system proposed here has been primarily built upon feature extraction. Where in total 60 features are taken into consideration[1]. The baseline experimental setup constitutes:

a) Logistic Regression,

b) Support Vector Machine

c) Convolutional Neural Network

d) Parallel Convolutional Highway Network

In the following model wordembeddings learned from a large corpus is used.The corpus consists of the data collected from Reddit, Facebook, Twitter, and keeping the hyperparameters constant, the word embeddings are then fed to convolutional neural network [2]. The model here achieves higher accuracy without any feature extraction or hyperparameter tuning. A simple CNN with one layer of convolution is used here.

This paper proposes the approach based on machine learning for detection of Thai clickbait. The clickbait messages often adopt eye-catching on wording, lagging of information on a content to attract visitors. We contribute the clickbait corpus by crowdsourcing, 30,000 of headlines are selected to draw up the dataset [3]. In this work attempt to develop clickbait detection model using two type of features in the embedding layer and three different of networks in the hidden layer.

Bi-LSTM with word levelembedding performs very well achieving accuracy rate of 0.98, f1-score of 0.98.

Clickbait spread wider and wider, with the development of online advertisements. It dissatisfies users because the article content does not go along with their expectation. Thus, recently clickbait detection has attracted more and more attention. Because of the limited information in headlines [4] traditional clickbait-detection methods rely on heavy feature engineering and fail to distinguish clickbait from normal headlines precisely. A convolutional neural network is useful for clickbait detection, since it uses pretrainedWord2Vec embeddings to understand the headlines semantically, and using different kernels finds various characteristic feature of the headlines.

However, different types of articles use different ways to gain users attention, and a pretrainedWord2Vec model cannot distinguish them easily. To address theseissues, a clickbait convolutional neural network (CBCNN) is built to consider not just the overall characteristics but also specific characteristics features from different types of article.

The results show that the method currently outperforms all the traditional clickbait detection methods and the Text-CNN model in terms of precision, recall and accuracy.

The use of misleading techniques in user-generatednews portalsare ubiquitous. Unscrupulous uploaders intentionally mislabel video descriptions aiming at increasing theirviews and results in increasing their ad revenue. This problem, usually termed as "clickbait," may severely undermine user experience [5].

In this work, study of the clickbait problem on is done on YouTubeby collecting metadata for 206k videos. To generate the solution, a deep learning model based on variational autoencoderssupporting the diverse modalities of data that videos include is devised.The proposed model relies on a limited amount of data because it is manually generatedlabelled data to classify a large corpus of unlabeled data. Theevaluation interprets that the proposed model offers improvedperformance when compared to other conventional models.

The analysis of the collected data shows that YouTube recommendation engine does not take into consideration the clickbait problem.

Thus, it allows recommending misleading videos to users.

III. Problem Statement

The work proposed in this paper addressses the following issues:

1) To identify Clickbait and non-Clickbait headline and classify them successfully.

2) To obtain the word embeddings for the words present in the dataset and specifically for rare words, which was found to be the common drawback in almost all proposed solutions till now.

3) To allow the system to run on CPU rather than GPU.

Thus, it is considered to be a binary classification problem, where the headline is taken into consideration.

IV. Data Collection And Visualization

The data is collected from various news sites.

The spam headlines are collected from sites such as ‘BuzzFeed’, ‘Upworthy’, ‘Viral Nova’, ‘That scoop’, ‘Scoop whoop’ and ‘Viral Stories’.

The relevant or non-spam headlines are collected from many trustworthy news sites such as ‘WikiNews’, ’New York Times’, ‘The Guardian’, and ‘The Hindu’.

The dataset in total has 32002 rows and 2 columns.

It has two columns.

The first column contains headlines and the second column has numerical labels or binary labels of clickbait in which 1 represents that it is clickbait and 0 represents that it is non-clickbait headline.

The dataset contains total 32000 rows of which 50% are clickbait and other 50% are non-clickbait, shows that dataset is equally distributed.

Table 1:Statistics of Dataset used

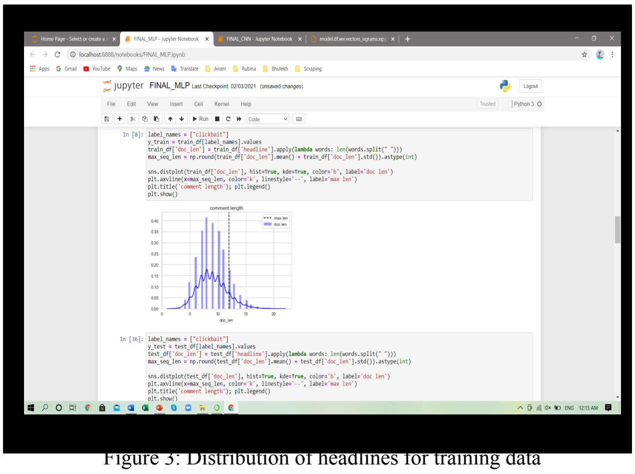
Total Headlines	Click-bait Headlines	Non-Click-bait Headlines	Vocabulary Length	Year of Creation
32,000	15,999	16,001	18,966	2015

The dataset is taken from [6] for our project.

The Dataset is then divided into 80:20 training and testing data respectively.

Table 2:Dataset Splitting for training and testing

Number of training data	25600
Number of testing data	6400

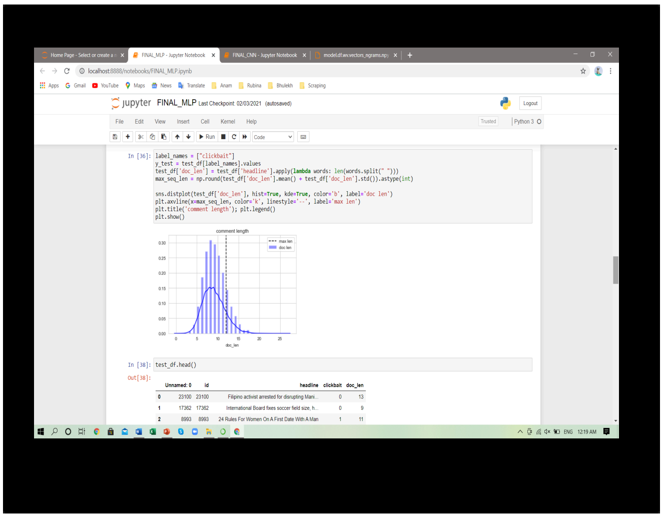


```
In [8]: label_names = ["clickbait"]
        y_train = train_df[label_names].values
        train_df['doc_len'] = train_df['headline'].apply(lambda words: len(words.split(" ")))
        max_seq_len = np.round(train_df['doc_len'].mean() + train_df['doc_len'].std()).astype(int)

        sns.distplot(train_df['doc_len'], hist=True, kde=True, color='b', label='doc len')
        plt.axvline(x=max_seq_len, color='k', linestyle='--', label='max len')
        plt.title('comment length'); plt.legend()
        plt.show()
```

FINAL_MLP - Jupyter Notebook
FINAL_CNN - Jupyter Notebook
model.df.wv.vectors_ngrams.npy
localhost:8888/notebooks/FINAL_MLP.ipynb
jupyter FINAL_MLP Last Checkpoint: 02/03/2021 (unsaved changes)
File Edit View Insert Cell Kernel Help
Run
Code
comment length
max len
doc len
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
doc_len

```
In [36]: label_names = ["clickbait"]
         y_test = test_df[label_names].values
         test_df['doc_len'] = test_df['headline'].apply(lambda words: len(words.split(" ")))
         max_seq_len = np.round(test_df['doc_len'].mean() + test_df['doc_len'].std()).astype(int)

         sns.distplot(test_df['doc_len'], hist=True, kde=True, color='b', label='doc len')
         plt.axvline(x=max_seq_len, color='k', linestyle='--', label='max len')
         plt.title('comment length'); plt.legend()
         plt.show()
```

Figure 3: Distribution of headlines for training data



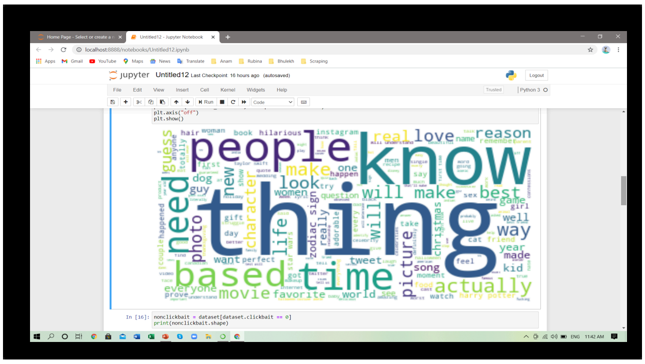
```
In [36]: label_names = ["clickbait"]
         y_test = test_df[label_names].values
         test_df['doc_len'] = test_df['headline'].apply(lambda words: len(words.split(" ")))
         max_seq_len = np.round(test_df['doc_len'].mean() + test_df['doc_len'].std()).astype(int)

         sns.distplot(test_df['doc_len'], hist=True, kde=True, color='b', label='doc len')
         plt.axvline(x=max_seq_len, color='k', linestyle='--', label='max len')
         plt.title('comment length'); plt.legend()
         plt.show()
```

FINAL_MLP - Jupyter Notebook
FINAL_CNN - Jupyter Notebook
model.df.wv.vectors_ngrams.npy
localhost:8888/notebooks/FINAL_MLP.ipynb
jupyter FINAL_MLP Last Checkpoint: 02/03/2021 (autosaved)
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
comment length
max len
doc len
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
25
doc_len

```
In [38]: test_df.head()

Out[38]:
     Unnamed: 0     id                                      headline  clickbait  doc_len
  0       23100  23100  Filipino activist arrested for disrupting Mani...          0       13
  1       17362  17362  International Board fixes soccer field size, h...          0        9
  2        8993   8993        24 Rules For Women On A First Date With A Man          1       11
```

Figure 4: Distribution of headlines for testing data

An additional variable “document length” is used here which accounts for total words present in a headline, in order to understand the distribution of headlines in training and testing datasets.

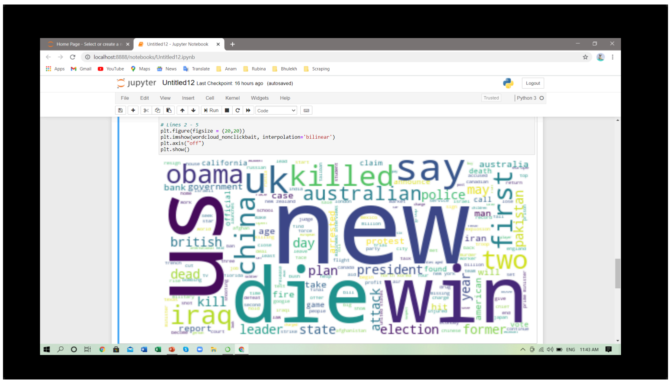


```
plt.axis("off")
plt.show()
```

Untitled12 - Jupyter Notebook
localhost:8888/notebooks/Untitled12.ipynb
jupyter Untitled12 Last Checkpoint: 16 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
people
know
thing
need
based
time
make
will make
look
love
real
reason
best
will
picture
actually
movie
favorite
world
photo
character
life
way
well

```
In [16]: nonclickbait = dataset[dataset.clickbait == 0]
         print(nonclickbait.shape)
```

Figure 5:Word Cloud for Clickbait Headlines



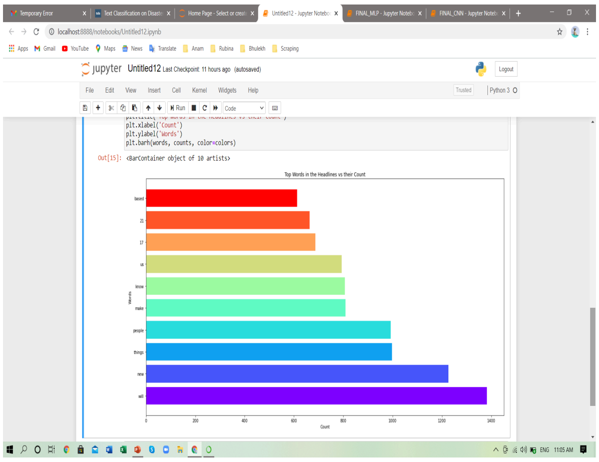
```
# Lines 2 - 5
plt.figure(figsize = (20,20))
plt.imshow(wordcloud_nonclickbait, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Untitled12 - Jupyter Notebook
localhost:8888/notebooks/Untitled12.ipynb
jupyter Untitled12 Last Checkpoint: 16 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
us
new
die
win
obama
uk
killed
say
australian
police
china
british
day
first
two
iraq
president
plan
leader
state
election
former

Figure 6: Word Cloud for NON-Clickbait Headlines

The above visualization of clickbait and non-clickbait datasets clearly shows the dissimilarities between these two. Furthermore, top words have also been identified that are present in a clickbait headline.

The following figure shows the top words and their count.



```
plt.xlabel('Count')
plt.ylabel('Words')
plt.barh(words, counts, color=colors)
```

```
Out[15]: <BarContainer object of 10 artists>
```

localhost:8888/notebooks/Untitled12.ipynb
jupyter Untitled12 Last Checkpoint: 11 hours ago (autosaved)
Top Words in the Headlines vs their Count
based
21
17
us
know
make
people
things
new
will
Words
0
200
400
600
800
1000
1200
1400
Count

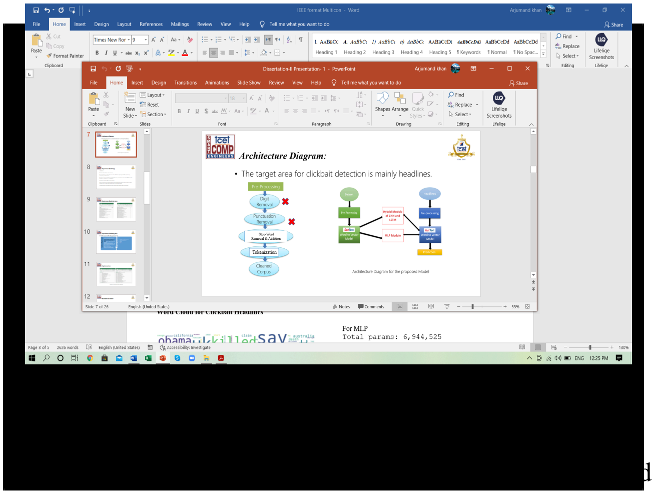
Figure 7: Words and their Count in Headlines

V. Proposed System

The entire process of the proposed system has been classified into phases for ease of operations.

A) The first phase consists of pre-processing of textual data using Natural Language Processing technique.

In this, extra white spaces, punctuations have been removed. Entire headline is converted from uppercase to lowercase. Stop Words Removal and tokenization have been done. While performing visualization on the dataset it has observed that numbers also play an important role in such clickbait headlines, hence numbers are kept for detailed analysis of headlines. On the other hand, stemming and lemmatization chop off the entire word in one way or other, so there will be nothing in specific to feed to a neural network, hence both these techniques are not used in here.



Dissertation-II Presentation- 1 - PowerPoint
Architecture Diagram:
Pre-Processing
Digit Removal
Punctuation Removal
Stop-Word Removal & Addition
Tokenization
Cleaned Corpus
Dataset
Pre-Processing
fastText Word to Vector Model
Hybrid Module of CNN and LSTM
MLP Module
Headlines
Pre-processing
fastText Word to Vector Model
Prediction
Architecture Diagram for the proposed Model
Word Cloud for Clickbait Headlines

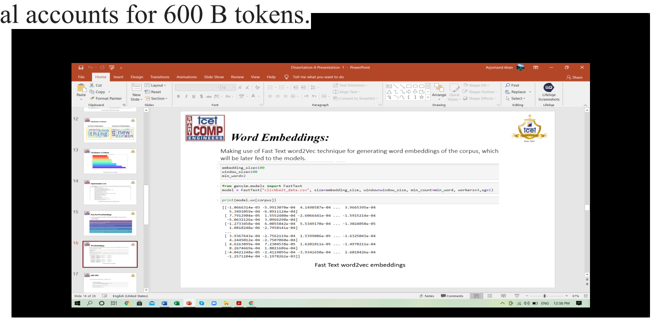
- The target area for clickbait detection is mainly headlines.

For MLP
Total params: 6,944,525

d for the dataset that will be fed to the neural network.

FastText: It allows users to learn and use text representations and text classifiers as It is an open-source, free, lightweight library. Standard, generic hardware is needed for working with FastText. Models can later be reduced in size and can be made able to fit on mobile devices. It is another word embedding method and an extension of the word2vec model. Instead of learning word vectorsdirectly, FastText represents each word as an n-gram of characters. This helps in capturing the meaning of shorter words and allowing the embeddings to understand suffixes and prefixes. It works on CPU rather than GPU. This makes our model really effective in terms of cost.

Here, for this model FastText embeddings provided by Facebook for text classification is used, which is 2-million-word vectors generated from common crawl in total accounts for 600 B tokens.



Making use of Fast Text word2Vec technique for generating word embeddings of the corpus, which will be later fed to the models.

```
embedding_size=100
window_size=100
min_word=2
```

```
from gensim.models import FastText
model = FastText("clickbait_data.csv", size=embedding_size, window=window_size, min_count=min_word, workers=4,sg=1)
```

```
print(model.wv[corpus])
```

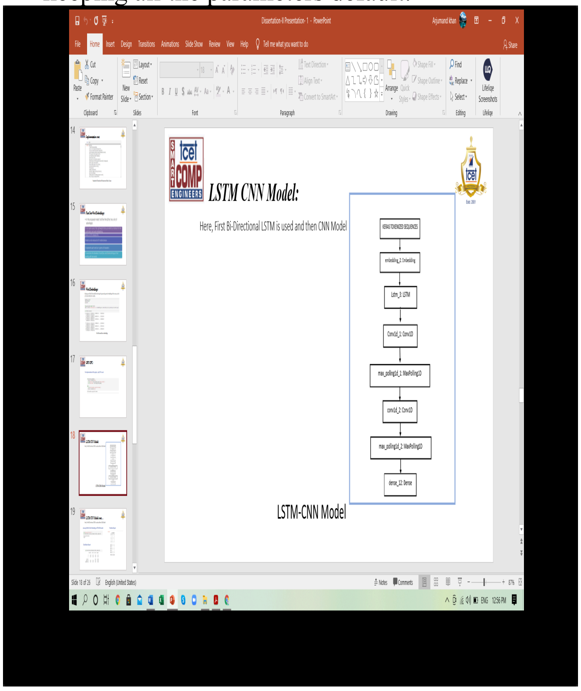
```
[[-1.0666314e-03 -5.9913070e-04  4.1498587e-04 ...  3.9665395e-04
   5.3491059e-04 -6.8911124e-04]
 [ 7.7912904e-05  1.5552480e-04 -2.6066441e-04 ... -1.5915214e-04
  -5.0632126e-04  3.0966298e-04]
 [-1.2733458e-04  6.0855842e-04  5.5349170e-04 ... -1.3024058e-05
   1.0818248e-06 -2.7958141e-04]
 ...
 [ 3.9367643e-04 -2.7562119e-04  1.5399086e-05 ... -1.6325043e-04
   4.2445012e-04 -2.7507060e-04]
 [ 4.6163899e-04  7.2304538e-05  1.6201811e-05 ... -1.4978211e-04
   8.2674669e-04  1.8021686e-04]
 [-4.0421248e-05 -2.4119095e-04 -3.9341650e-04 ...  2.6018426e-04
  -1.2571104e-04 -1.1978262e-03]]
```

Fast Text word2vec embeddings

Figure 9: Fast Text word2vec embeddings

The next and the third phase is model building with the embeddings generated.

B) Hybrid Bi-Directional LSTM-CNN model: Flowchart for the model is shown in the following figure. The FastText word embeddings will be first fed to bi-directional LSTM and later to CNN, keeping all the parameters default.



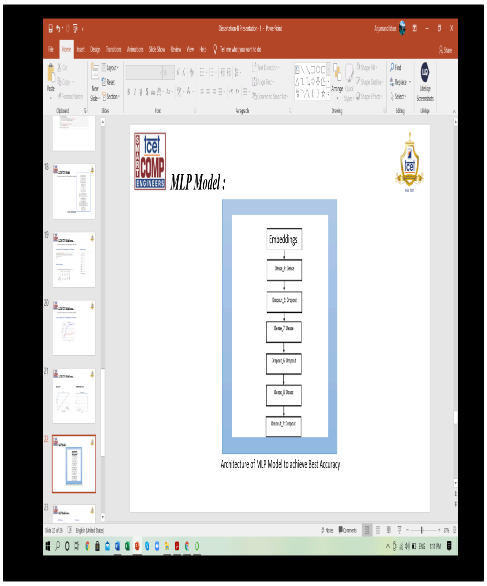
Dissertation-II Presentation- 1 - PowerPoint
LSTM CNN Model:
KERAS TOKENIZED SEQUENCES
embedding_2: Embedding
Lstm_3: LSTM
Conv1d_1: Conv1D
max_polling1d_1: MaxPolling1D
conv1d_2: Conv1D
max_polling1d_2: MaxPolling1D
dense_12: Dense
LSTM-CNN Model

Here, First Bi-Directional LSTM is used and then CNN Model

Figure 10: Bi- Directional LSTM-CNN Hybrid Model

The loss function used here is binary cross entropy and Adam optimizer is used along with "sigmoid" activation function.

C) Multi- Layer Perceptron Model: In machine learning MLP algorithm is known to be the backbone of deep learning. Using MLP with proper parameters and FastText word embeddings, it can give considerable result.



MLP Model :

Architecture of MLP Model to achieve Best Accuracy

Figure 11: Architecture of MLP Model to achieve Best Accuracy

The parameters here are kept constant also same optimization, loss and activation functions have been used as for LSTM-CNN model to achieve the best accuracy.

VI. Model Evaluation and Result

The last and most crucial phase comprises of model evaluation and discovering really considerable results.

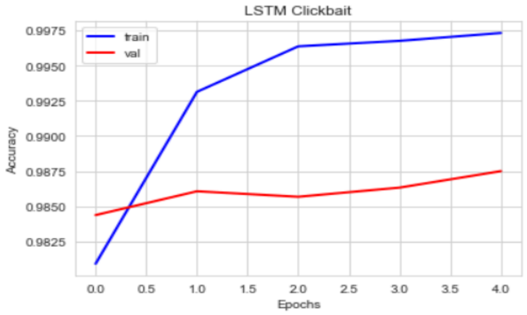
Table 3: For Bi-Directional LSTM-CNN

Total parameters	7,106,113
Trainable parameters	165,313

Table 4: For MLP

Total parameters	6,944,525
Trainable parameters	3,725

It can be clearly seen that for both, Bi-Directional LSTM-CNN and MLP count of Trainable parameters is much lower compared to total parameters clearly states that these are optimal weights found by the model to reduce the cost function of the model.



From the above given graphs and performance metrics it is evident that MLP model is doing good and can be

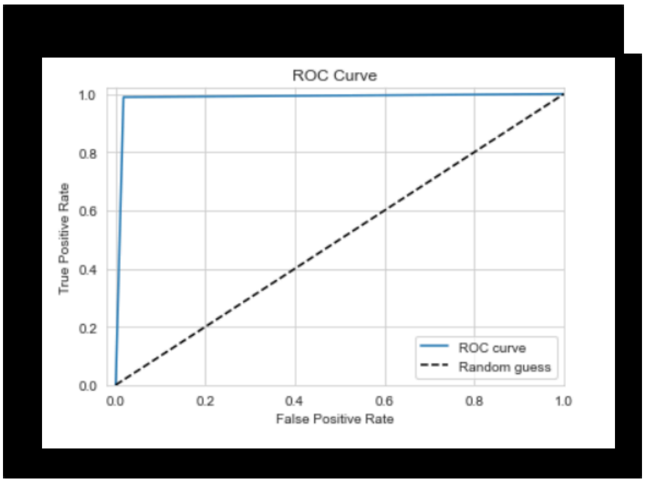


Figure 13: ROC Curve with FastText Word Embeddings on Bi-Directional LSTM-CNN model

A Receiver Operating Characteristic (ROC) curve illustrates the diagnostic ability of a binary classifier. The closer is the curve to top-left corner greater is the accuracy of the classifier. In our case it is showing remarkable accuracy.

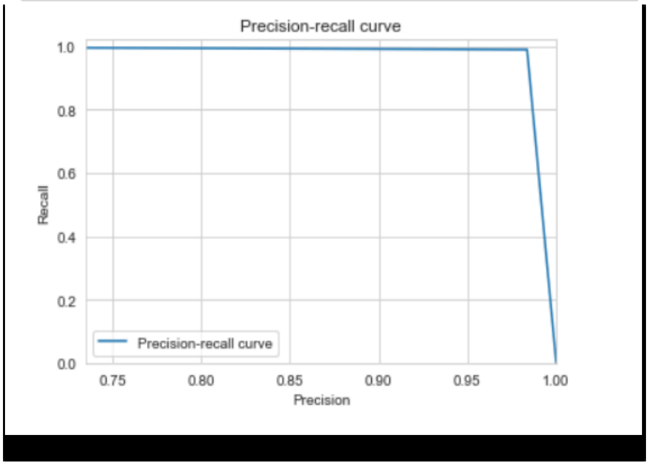


Figure 14: Precision-Recall curve with FastText Word Embeddings on Bi-Directional LSTM-CNN model

The curve denotes an accurate classifier, as it shows high precision denoting a low false positive rate and recall denoting a low false negative rate.

Table 6: Analysis of MLP model using performance metric

	Accuracy	Precision	Recall	F1-Score
Click-bait	86.00	91.00	80.00	85.00
Non-Click-bait		82.00	92.00	87.00

themselves and flooding the online portals with fake news, novel neural network architectures are been proposed, and it has been found out that word embeddings play a major role in increasing the accuracy and reliability of the system. Also, the Bi-Directional LSTM-CNN model performs better than MLP model with FastText word embeddings and outperforms all other classification algorithms and Word2Vec techniques by achieving highest accuracy till now. However, the job is far from over so, the future scope includes not only detecting click-baits but also blocking them.

VII. Acknowledgment

I gratefully acknowledge the support, guidance and encouragementof my Dissertation Guide Associate Professor Dr. Megharani Patil ma'am for this novel work.

References

[1] Peter Adelson, Sho Arora, and Jeff Hara, "Clickbait; Didn't Read: Clickbait Detection using Parallel Neural Networks", 2018.

[2] Amol Agrawal, "Clickbait Detection using Deep Learning", 2nd International Conference on Next Generation Computing Technologies,2016.

[3] PraphanKlairith, SansiriTanachutiwat, "Thai Clickbait Detection Algorithms using Natural Language Processing with Machine Learning Techniques", International Conference on Engineering, Applied Sciences, and Technology (ICEAST), 2018.

[4] Hai-Tao Zheng, Jin-Yuan Chen, Xin Yao, Arun Kumar Sangaiah, Yong Jiang, Cong-Zhi Zhao, "Clickbait Convolutional Neural Network",2018.

[5]SavvasZannettou, SotiriosChatzis, KostantinosPapadamou, Michael Sirivianos, "The Good, the Bad and the Bait: Detecting and Characterizing Clickbait on YouTube",IEEE Symposium on Security and Privacy Workshops, 2018.

[6]Saumya Pandey, Gagandeep Kaur, "Curious to Click It? -Identifying Clickbait using DeepLearning and EvolutionaryAlgorithm", International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018.

[7] Sawinder Kaur, Parteek Kumar, PonnurangamKumaragurub, " Detecting clickbaits using two-phase hybrid CNN-LSTM biterm model", Journal of Expert Systems with Applications, 2020.

[8] Lanyu Shang, Daniel Zhang, Michael Wang, Shuyue Lai, Dong Wang, "Towards Reliable Online Clickbait Video Detection:A Content-Agnostic Approach",Journal of Knowledge Based Systems, 2019.

[9]Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, NiloyGanguly, "Stop Clickbait: Detecting and Preventing Clickbaits in Online News Media", IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2016.

[10] Philogene Kyle Dimpas, Royce Vincent Po, Mary Jane Sabellano "Filipino and English Clickbait Detection Using a Long Short Term Memory Recurrent Neural Network", 2017.

[11] Vaibhav Kumar, Dhruv Khattar, Siddhartha Gairola "Identifying Clickbait: A Multi-Strategy Approach Using NeuralNetworks", 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, 2018.

[12] Kai Shu, Suhang Wang, Thai Le, Dongwon Lee,Huan Li,"Deep Headline Generation for Clickbait Detection", IEEE International Conference on Data Mining, 2018.