Novel Technique: Detecting Clickbait Using Machine Learning and Deep Learning Model

Yee Zhian Liew, Saurabh Aggarwal, Abhishek Prabhudesai, Sowndhariya Nandarajkumar
Chandrasekar Vuppalapati
Department of Software Engineering
San Jose State University, USA

Abstract—As far as we know that bait means luring someone to do something. Click-baiting is to attracts readers attention to click it. Click-baiting is a growing phenomenon on the internet, and it is defined as a method of exploiting cognitive biases to attract online viewership, that is, to attract "clicks." [1] In other words, these headlines contain text which leaves the reader curious about what the article contents might be, or they contain text about topics not really covered in the article itself. In order to avoid readers from clicking it, we attempt to detect the clickbait with our design model. In this paper, we are going to use 2 different datasets to understand the extent of clickbait practice, its impact and user engagement by using our own developed clickbait detection model. Moreover, we study the clickbait problem on YouTube by collecting metadata of YouTube videos. To address it, we revise a deep learning model based on variational autoencoders that supports the diverse modalities of data that videos include. The proposed model relies on a limited amount of manually labeled data to classify a large corpus of unlabeled data. The model uses LSTM model embeddings learned from a large corpus. The accuracy of the model is *75%*. In addition, as opposed to previous studies, we found that clickbait post website has more images compare to non-clickbait**.**

Index Terms **- clickbait, tokenize, NLTK, YouTube, Deep Learning, LSTM, Machine Learning**

I. INTRODUCTION

THE basic trick of how the publisher gets reader's attention is creating a very interesting headlines. [2] Without curiosity, readers will not border to see the content of the web. However, this bait often has a very bad content and leaves the readers disappointed. According to Cambridge Dictionary, clickbait is define as articles, photographs which is on the internet that are intended to attract readers' attention and encourage readers to click on links to particular websites. [2] Example of the common clickbait title will be *"Top 10 tricks will change your life", "5 things that will change your life!!" and "She said she had HIV: What happens next will blow your mind!"*. [3] In order to detect click-bait, a supervising learning model can be created. Before creating a model, we need to have a decent amount of dataset. We collected extensive dataset of clickbait and non-clickbait from GitHub which contains 16,000 headlines.

Besides, social media such as YouTube will contains clickbait headlines. Chances for a reader to click on clickbait header without noticing it is high. [4] Seldom we ask ourselves, why some of the YouTube users trying so hard to post interesting headlines which is not related to their content? The answer is simple: each of them trying to get reader's attention which is views, likes, and shares. We refer these eye-catching thumbnails technique as clickbait. [5]

II. KDD PROCESS

The term *Knowledge Discovery in Databases*, or KDD for short, refers to the broad process of finding knowledge in data, and emphasizes the "high-level" application of particular data mining methods. It is of interest to researchers in machine learning pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, and data visualization. [6]

Fig.1. Basic KDD process

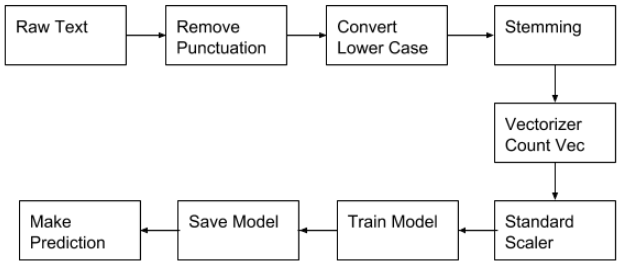

Fig.2. Sentiment Analysis Process Flow

III. DATASET

A. **Clickbait Dataset:** *The clickbait corpus consists of article headlines from 'BuzzFeed', 'Upworthy', 'ViralNova'. [7]*

B. **Non clickbait Dataset:** *The non-clickbait article headlines are collected from 'WikiNews', 'New York Times', 'The CNN news'. [7]*

IV. FORMULA

A. **Stop word:** According to our result, we found out that click-bait used more stop word. For example: a, an, but, how, the, etc. NLTK has the stop word package which is very helpful to filter out the stop word from texts. [8]

B. **No of character/Length of headlines:** Readers always wanted to know what the article content is roughly about so the first thing they will read through is the headline. From what we have found, click-bait seems to have a longer length of headline compare to non-click-bait.

$$charnum(x) = \{len_{char}(x),\ if\ \ni x \quad -1\ ,\ otherwise\}$$

C. **Word length /difference between no of words:** For standard or formal headlines from the news, it will use **you are** instead of **you're**. Clickbait headline tend to have a shorter word. [9]

D. **Punctuations:** Punctuations seems very rare to occur on standard headlines. According to our model, **c**lickbait headline uses informal punctuation patterns very frequently. Punctuations such as !?, ..., ***! One famous clickbait example will be: "*A 5 years old girl is left alone: You won't believe what's happening next!*" [9]

E. **Common clickbait phases/common words:** As the result, there are some common phases are occurred very frequently. For example: top 10, "will blow your mind", "you would not believe it". [9]

$$NumCommonWords(cont_x) = key(article(p)) \cap words(cont_x)$$

F. **Formal and informal English word:** Formal or trustable news headline basically contains formal words to gain readers' trust. Clickbait headline is easier to detect by for example: clickbait usually contains words such as blow up instead of exploding, point out vs indicates and ring up vs call.

$$Formal - words(x) = \frac{lang-dict_{formal}(words(x))}{words(x)}$$

$$Informal - words(x) = \frac{lang-dict_{informal}(words(x))}{words(x)}$$

V. TFIDF

Term frequency-inverse document frequency or text feature extraction is a statistic calculation on pinpointing how important a word is to a document. In another way it is using to judge the topic of an article by the words it contains. Relevance is the key measurement for TF-IDF but not frequency. First, TF calculates the total number of the same word that appears in a document. Since there will be word which is not important such as "and", "or", they are systematically discounted and that is what IDF does. TF-IDF intend to keep the important words for analyzing purpose and remove words that seems to be useless.

$$W_{i,j} \times log \frac{N}{df_i}$$

$$tf_{i,j} = number\ of\ occurrences\ of\ i\ in\ j$$

$$df_i = number\ of\ documents\ containing\ i$$

$$N = total\ number\ of\ documents$$

Fig.3. Mathematical Calculation for TFIDF

No doubt that this term is often used in text mining because it can extract each word as a string for easier calculation purposes. TFIDF vectorizer is used to transforms text to feature vectors that can be used as input to estimator. We can analyze it 4 for 2 ways: character and word. Basically, it will be comparing each characters or word in the document and find out which is appearing the most.

VI. YOUTUBE DATASET

For YouTube dataset, we collect the metadata from 2010 to 2017. With the small detail included as comments from users, tags, views, likes, shares, thumbnails, we can implement heat graph analysis on how strong on each detail are related to the headlines. [10] Before starting to analyze, we need to apply normalized factor in order to remove the bias towards click baits.

VII. FORMULA

In order to understand well on how to detect clickbait from YouTube, we will start giving brief analysis on each component.

A.**Headline**: The headline is the most influential text component. It must always have very interesting keywords to catch readers to click on it. [5] Youtubers usually love to use exaggerating phrase such as "viral" and "epic". Often readers will get disappointed or frustrated because the content is not even related to the headlines. For clickbait, we have found out that most of the headline uses the word: "remember" as the first word of the headlines. For example: *"Remember this celebrity? Well they don't look like this anymore."* Moreover, the second most word is used in clickbait YouTube headline is *"hot"* ,*"sexi"*, and money. [10] For example: "How to make $4000 at home in 2 hours?" This is the advertisement or various opportunity blog to get customers.

B. **Thumbnail:** Thumbnail is defined as the book cover of any online video. In short, thumbnail is the striking or appealing image that will catch reader's attention. [1] Often Youtuber will find the most attractive image of their entire video. Discarding the blurry and boring image as it will not ever get attention from the viewers. For research, 80% of the clickbait image will contains "pretty" or "hot" images. Once you click on it, the content is exactly different with the "image" the Youtuber is showing. [5]

C. **Tag**: For easy understanding, tags are identical to hashtag where you can find in social media such as Instagram or twitter. Youtubers are using the funny or impressive tag to lure more readers to search it. We noticed that some most common words that used in clickbait video are: "impossible" or "unbelievable" phrase that sounds fake.

D. **Comment**: Comment is a written type for expressing an opinion or thinking of someone. Usually people who is commenting are the victims who had already falls into clickbait video trap. We can detect whether the video is clickbait from the comment that people post under the video. Basically, You Tube's comment has a "flag" button to raise the awareness to YouTube community about the problem. As a result, frequency of the 'flag' is raised can be used to determine the video as clickbait. [10]

VIII. REVISITING THE PROBLEM

From what we have learn that, detecting a clickbait just from the header is not accurate enough. Imagine after you define a headline as a clickbait based on our model. [11] But the content fits perfectly with the headlines. It is describing everything which is related to the headline. In order to improve our model, we need to evaluate the content before declaring that it is a clickbait headline. From our result, we found out that most of the clickbait websites have more links compared to the non-clickbait websites. [11] In addition, clickbait websites will have more images than non-clickbait websites. Commonly used word for clickbait are: 'things', 'people', 'will', 'know'. [11]

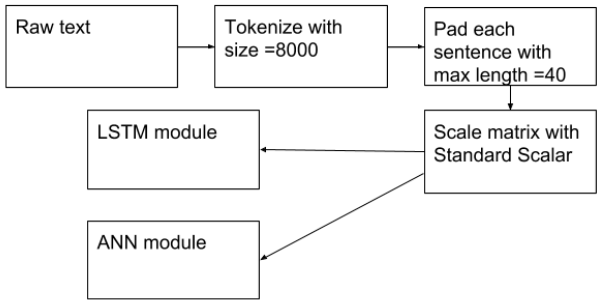

Fig.4. Deep Learning with Text

IX. DEEP LEARNING

Before long short-term memory is introduced, Recurrent neural network is the machine learning that used to solve for classification and regression problem. [12] Basically, it contains memory to store data which is important and remove which is not. It makes the decision by connecting several neural networks together and compute a desire output. Unfortunately, the two main problem that RNN is suffering are vanishing gradient and exploring gradient. [13] This is costing RNN to be unstable with the decision making and that is LSTM is invented.

LSTM is the unit of RNN which has the capability to store the information in long term. Internally, it contains states cells that act as a long term and short-term memory. Basically, LSTM network stores the information in a loop and has the ability control the flow of the information to flow from one block to another. LSTM working mechanism is achieving almost on human level. For example: In text, human is understanding the sentences by reading it word by word. [13] Same as LSTM, they are scanning through the first word and predict the next word by past experience.

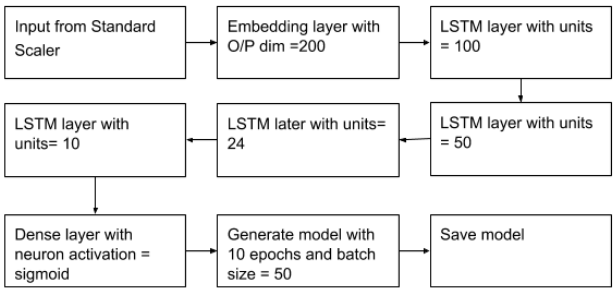

Fig.5. Structure of LSTM Model

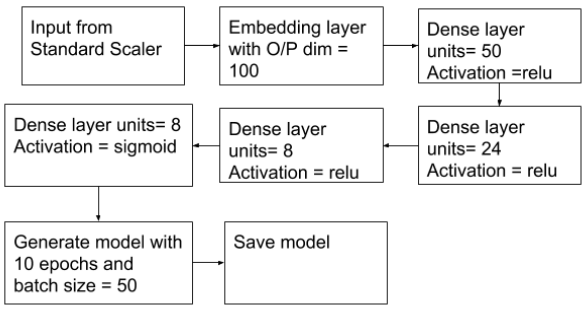

Fig.6. Structure of ANN Model

X. CALCULATION

Logistic Regression: It is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal or ratio-level independent variables. We can calculate the **ROC AUC, Precision, Recall and F1 score**. Receiver operating characteristic curve (ROC) is a graph showing how accurate or precise is the model for classification problem. [3] The higher the score which means the better the accuracy of our model. ROC can use confusion matrix to determine how good your ROC curves for your model.

Basically, there are 4 main parameters which are True/False positive rate and True/False negative rate. True positive rate is usually measuring the sensitivity or probability of detection. It measures the probability of how many true

components are identified correctly. However, false positive rate is used to measure the probability of false positive component which falls into the true positive. It will be vice versa for the other 2 parameters.

Area Under the curve (AUC) is a statistic calculation which shows us how well our model on prediction. [3] We can apply AUC on different model and whichever model has the AUC can be seen as the best model.

Moreover, precision score is to calculate how precise the true positive is. High precision often leads to low false positive rates. For example: a bunch of words are labeled as positive words but how many actually is label correctly. In addition, recall score is calculating how many of the actual positive our model capture through labeling it as positive. F1 score is calculating the average between precision and recall. F1 score is more useful than accuracy in general.

$$TPR = \frac{TP}{P} = \frac{TP}{TP+FN} = 1 - FNR = \frac{\Sigma\, True\ positive}{\Sigma\, Condition\ positive}$$

$$TNR = \frac{TN}{N} = \frac{TN}{TN+FP} = 1 - FPR = \frac{\Sigma\, True\ negative}{\Sigma\, Condition\ negative}$$

$$F1\ Score = \frac{2\times(Recall \times Precision)}{(Recall + Precision)}$$

$$Accuracy = \frac{\Sigma\, True\ positive + \Sigma\, True\ negative}{\Sigma\, Total\ population}$$

Fig.7. Mathematical Calculation for TPR, TNR, F1 and Accuracy

XI. ML ALGORITHM

A. **Decision Tree:** A decision tree will contain leaves and nodes which branch out. The internal nodes are tests and whose leaf nodes are categories. Basically, each of the attributes will be tested on each internal node. [12] Then, each branch from the internal node will select the best attribute. Best attribute is selected according to maximum information that can be extracted, known as information ratio.

```
generateDecisionTreeClassifier (X_train,y_train)
y_pred_DT = predictDecisionTreeClassifier (X_test)
cm_DT = generateConfusionMatrix (y_test,y_pred_DT)
cm_DT
```

```
array([[713,  77],
       [115, 510]], dtype=int64)
```

Fig.8. Confusion Matrix of DT

	precision	recall	f1-score	support
0.0	0.86	0.89	0.87	790
1.0	0.85	0.82	0.83	625
avg / total	0.85	0.86	0.85	1415

Fig.9. Classification Report of DT

According to the Decision Tree confusion matrix as shown above, the model predicted 713 correctly as True positive whereas predicted 77 wrong as False positive. Also, the model predicted 510 as True negative but it predicted 115 wrong as False negative.

B. **K-nearest Neighbor:** Often KNN algorithm is used for both classification and regression problems. Variety of applications such as finance, text classification, image recognition is using KNN to solve their complicated task. Feature similarity is the approach that KNN used. Indeed, KNN is a lazy learning algorithm which means that no training data is required for generation model. This is a big advantage because the training time will be way faster. [12] The disadvantages of KNN is that it needs to scan all the necessary data point before it starts to test. More time leads to more memory and storage. Setting K values is the hardest decision making for KNN. Number of K values is number of clusters going to be formed. For example: when K =1, a random data point is selected, and it will calculate the distance and find the nearest data point and cluster together as the same label.

```
generateKNNClassifier (X_train,y_train)
y_pred_KNN = predictKNNClassifier (X_test)
cm_KNN = generateConfusionMatrix (y_test,y_pred_KNN)
cm_KNN
```

```
array([[465, 325],
       [218, 407]], dtype=int64)
```

Fig.10. Confusion Matrix of KNN

	precision	recall	f1-score	support
0.0	0.76	0.37	0.50	790
1.0	0.52	0.85	0.64	625
avg / total	0.65	0.58	0.56	1415

Fig.11. Classification Report of KNN

According to the KNN confusion matrix as shown above, the model predicted 462 correctly as True positive whereas predicted 325 wrong as False positive. Also, the model predicted 407 as True negative but it predicted 218 wrong as False negative.

C. **Random Forest:** This algorithm is an ensemble learning method for classification, regression that operate by constructing multiple decision tree. For example: creating few random subsets for the problem statement. For each subset, it will design as each of the decision tree. Assuming there is a new subset need to be classify and the new subset will be throw into the decision trees.[12] Each decision trees will provide an output and the highest output will be as the new subset class.

```
generateRandomForestClassifier (X_train,y_train)
y_pred_RF = predictRandomForestClassifier (X_test)
cm_RF = generateConfusionMatrix (y_test,y_pred_RF)
cm_RF
```

```
array([[757,  33],
       [124, 501]], dtype=int64)
```

Fig.12. Confusion Matrix of RF

	precision	recall	f1-score	support
0.0	0.86	0.93	0.90	790
1.0	0.91	0.81	0.85	625
avg / total	0.88	0.88	0.88	1415

Fig.13. Classification Report of RF

According to the Random Forest confusion matrix as shown above, the model predicted 757 correctly as True positive whereas predicted 33 wrong as False positive. Also, the model predicted 501 as True negative but it predicted 124 wrong as False negative.

D. **Support Vector Machine:** A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. [12] In other words, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples. For instance, a hyperplane is a line that is dividing 2 different class of objects in two parts. This method only can be used in 2 dimensional for classification.

```
generateSVMClassifier (X_train,y_train)
y_pred_SVM = predictSVMClassifier (X_test)
cm_SVM = generateConfusionMatrix (y_test,y_pred_SVM)
cm_SVM
```

```
array([[712,  78],
       [ 65, 560]], dtype=int64)
```

Fig.14. Confusion Matrix of SVM

	precision	recall	f1-score	support
0.0	0.93	0.90	0.91	790
1.0	0.88	0.91	0.89	625
avg / total	0.90	0.90	0.90	1415

Fig.15. Classification Report of SVM

According to the Decision Tree confusion matrix as shown above, the model predicted 712 correctly as True positive whereas predicted 78 wrong as False positive. Also, the model predicted 560 as True negative but it predicted 65 wrong as False negative.

E. **Naive Bayes:** Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.. It is one of the most effective algorithms comparing with others classification algorithms. Basically, this algorithm can classify a word whether it is positive, negative or neutral. For instance, it will convert the document into attribute set where columns is going to be possible word and row is how many times the word appear in the document. According to the attribute set, it will define which word should falls on which categories.

```
generateNaiveBayesClassifier (X_train,y_train)
y_pred_NB = predictNaiveBayesClassifier (X_test)
cm_NB = generateConfusionMatrix (y_test,y_pred_NB)
cm_NB
```

```
array([[727,  63],
       [165, 460]], dtype=int64)
```

Fig.16. Confusion Matrix of NB

	precision	recall	f1-score	support
0.0	0.83	0.93	0.88	790
1.0	0.90	0.76	0.83	625
avg / total	0.86	0.86	0.86	1415

Fig.17. Classification Report of NB

According to the Naive Bayes confusion matrix as shown above, the model predicted 727 correctly as True positive whereas predicted 63 wrong as False positive. Also, the model predicted 460 as True negative but it predicted 165 wrong as False negative.

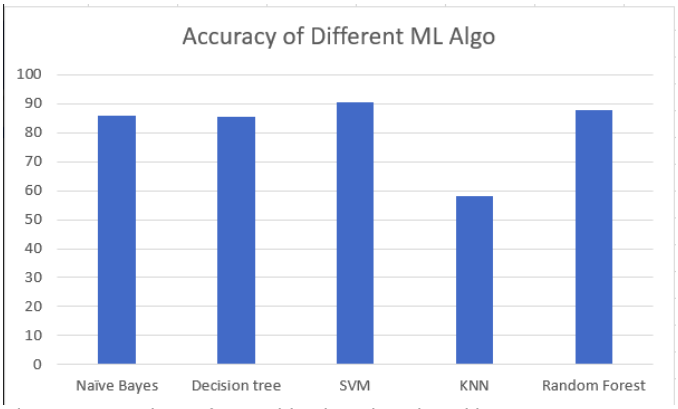

Fig.18. Comparison of 5 machine learning alogorithm

F. LSTM Model

	1	2	3	4	5	6	7	8	9	10
loss	0.63	0.58	0.55	0.55	0.52	0.53	0.52	0.51	0.50	0.50
accuracy	0.59	0.68	0.71	0.72	0.72	0.73	0.73	0.73	0.73	0.74

TABLE.1. LSTM Model with 10 Epochs

According to the LSTM model, we set our epochs as 10. The accuracy result we got is 79%. As you can see, the loss is keep decreasing on every epoch which and the accuracy is keep increasing.

G. ANN Model

	1	2	3	4	5	6	7	8	9	10
loss	0.69	0.67	0.61	0.55	0.58	0.57	0.56	0.55	0.50	0.55
accuracy	0.56	0.56	0.57	0.58	0.74	0.74	0.75	0.75	0.75	0.75

TABLE.2. ANN Model with 10 Epochs

According to the ANN model, we set our epochs as 10 too. The accuracy result we got is 74%. For this model, the loss is keep decreasing but it does not decrease as fast as LSTM model. This is causing the accuracy to be lower.

Model	Accuracy	Precision	Recall	F1-score
RF	0.88	0.88	0.88	0.88
DT	0.85	0.85	0.86	0.85
KNN	0.65	0.65	0.58	0.56
NB	0.86	0.86	0.86	0.86
SVM	0.90	0.90	0.90	0.90
LSTM	0.74	0.72	0.71	0.70
ANN	0.75	0.73	0.71	0.71

TABLE.3. Result from all baseline and Neural Network model

XII. Conclusion

In this paper, we have tested using traditional Machine learning algorithm and comparing with result. From Machine learning perspective, SVM gives the most accuracy which is 90%. The second comes to Naive Bayes and Random Forest which is 86% accuracy. The worst ML algorithm for our dataset is KNN which only have 56% accuracy. Furthermore, we have added features for our model because we wanted to test whether features from websites will affects our model accuracy or not. We have implemented deep learning LSTM and ANN model and it gives us lower accuracy which is approximately 75%, comparing to traditional ML algorithm.

XIII. References

[1] Wiegmann, M., Völske, M., Stein, B., Hagen, M. and Potthas, M. (2018). *Heuristic Feature Selection for Clickbait Detection*. [online] Arxiv.org. Available at: https://arxiv.org/pdf/1802.01191.pdf [Accessed 30 Nov. 2018].

[2] Elyashar, A., Bendahan, J. and Puzis, R. (2018). *Detecting Clickbait in Online Social Media: You Won't Believe How We Did It*. [online] Arxiv.org. Available at: https://arxiv.org/pdf/1710.06699.pdf [Accessed 30 Nov. 2018].

[3] DeGrave, K. (2018). *Project Pages - An Integrated Scientific Blogging Template*. [online] Degravek.github.io. Available at: https://degravek.github.io/project-pages/project1/2017/04/28/New-Notebook/ [Accessed 30 Nov. 2018].

[4] Xing, Y. (2018). *How does clickbait work: An eye-tracking method to discover people's reactions*? [online] Www-users.cs.york.ac.uk. Available at: https://www-users.cs.york.ac.uk/alistair/projects/yx1058.pdf [Accessed 30 Nov. 2018].

[5] Qu, J., Hißbach, A., Gollub, T. and Potthast, M. (2018). *Towards Crowdsourcing Clickbait Labels for YouTube Videos*. [online] Webis.de. Available at: https://webis.de/downloads/publications/papers/potthast_2018a.pdf [Accessed 30 Nov. 2018].

[6] DBD, U. (2018). *KDD Process/Overview*. [online] Www2.cs.uregina.ca. Available at: http://www2.cs.uregina.ca/~dbd/cs831/notes/kdd/1_kdd.html [Accessed 4 Dec. 2018].

[7] Uddin Rony, M., Hassan, N. and Yousuf, M. (2018). *Diving Deep into Clickbaits: Who Use Them to What Extents in Which Topics with What Effects?* [online] Arxiv.org. Available at: https://arxiv.org/pdf/1703.09400.pdf [Accessed 30 Nov. 2018].

[8] Woolf, M. (2018). *Visualizing Clusters of Clickbait Headlines Using Spark, Word2vec, and Plotly*. [online] minimaxir | Max Woolf's Blog. Available at: https://minimaxir.com/2016/08/clickbait-cluster/ [Accessed 30 Nov. 2018].

[9] Chakraborty, A., Paranjape, B., Kakarla, S. and Ganguly, N. (2018). *Stop Clickbait: Detecting and Preventing Clickbaits in Online News Media*. [online] Arxiv.org. Available at: https://arxiv.org/pdf/1610.09786.pdf [Accessed 30 Nov. 2018]

[10] Zannettou, S., Papadamou, K., Chatzis, S. and Sirivianos, M. (2018). The Good, the Bad and the Bait: Detecting and Characterizing Clickbait on YouTube. [online] Available at: https://www.researchgate.net/publication/323960601_The_Good_the_Bad_and_the_Bait_Detecting_and_Characterizing_Clickbait_on_YouTube/download [Accessed 30 Nov. 2018].

[11] Thakur, A.(2018).Clickbaits Revisited: Deep Learning on Title + Content Features to Tackle Clickbaits. [online] Available at: https://www.linkedin.com/pulse/clickbaits-revisited-deep-learning-title-content-features-thakur/ [Accessed 30 Nov. 2018].

[12] Cornn, K. (2018). *Clickbait article detection using deep learning: these results will shock you.* [online] Cs230.stanford.edu. Available at: http://cs230.stanford.edu/files_winter_2018/projects/6931206.pdf [Accessed 30 Nov. 2018].

[13] Shu, K., Wang, S., Le, T., Liu, H. and Lee, D. (2018). *Deep Headline Generation for Clickbait Detection*. [online] Public.asu.edu. Available at: http://www.public.asu.edu/~skai2/papers/clickbait_2018.pdf [Accessed 30 Nov. 2018].