Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 227 (2023) 282–290

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Conference on Computer Science and Computational Intelligence (ICCSCI 2023)

Tackling Clickbait with Machine Learning: A Comparative Study of Binary Classification Models for YouTube Title

Tora Sangputra Yopie Winarto[a,*], Kevin Wijaya[a], Muhammad Abdullah Faqih[a], Simeon Yuda Prasetyo[a], Yohan Muliono[a]

[a]*Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480*

Abstract

Clickbait is a form of internet content whose function is to attract attention and entice users to click on a link. Its main goal generally is to generate more advertisement revenue for the creator. Clickbait is featured heavily on all kinds of social media, especially YouTube where there is a strong financial incentive to do so. Clickbait content constitutes 47.56% of content from mainstream broadcast media and US companies spent an average of 9.8% of their advertising budget on clickbait contents. Clickbait classification is the first and most important step in resolving the proliferation of clickbait content. Contributing to this, we aimto detect YouTube clickbait videos by building several binary classification machine learning models trained on an open-sourceddataset of 31.987 English YouTube video titles from GitHubGist to differentiate between clickbait or non-clickbait YouTube titles.The machine learning models are based on Na¨ıve-Bayes, Support Vector Machine (SVM), and Long Short-Term Memory (LSTM)Network, with the final objective to compare each model's resulting effectiveness. The best-performing resulting model from thisstudy is a kernel TF-IDF SVM model scoring 98.53% on accuracy, precision, recall, and f1-score which outperforms the past experiments that is using the same models.

Keywords: clickbait; machine learning; Na¨ıve-Bayes; SVM; LSTM;

1. Introduction

Clickbait has been a constant and growing problem on the internet with various social media posts, news outlets, and other forms of media delivery [1]. It uses deceptive wordings, images, and such to entice people to view a certain content and/or believe a certain narrative. Generally, this practice is done to increase the amount of advertisement revenue that the creator will get from increased engagement. Clickbait content makes up around 47.56% of content from mainstream broadcast media [2]. This number is even higher in what's classed as ”unreliable media” such as

* Corresponding author. Tel.: +62-21-5369-6969 ; fax: +62-21-530-0244.
Email address: tora.winarto@binus.ac.id

10.1016/j.procs.2023.10.526

user generated content in social media because of a lack of content moderation and proof-reading that an official media might employ. From one analysis by media-research company Ebiquity, they found that their US clients spent an average of 9.8% of their advertising budgets on clickbait contents [3].

On YouTube, clickbaits are an exceptionally big problem compared to other social media [4]. This is because YouTube has a more lucrative per-click advertising revenue compared to other social media owing to the fact that it hosts longer format videos [5]. At its best, clickbaits can be a useful way for content creators to make people click on their media and receive more views. However, at its worst, it can be a way to make more advertisement money while making subpar videos and even as a tool for propaganda and misinformation [6]. From our observation forms of clickbait on YouTube can include:

- Titles and/or thumbnails that provoke the user into viewing the video,
- Video content that doesn’t match at all with its title and/or thumbnails,
- Non-factual titles and/or thumbnails.

Identifying clickbaits is an important part of reducing or stopping clickbait content from appearing on our social media timelines [7]. YouTube right now relies only on manual user flagging to identify and takedown clickbait content [4]. To that end, [8]. We envision that this model can be used to automate the filtering of clickbait YouTube videos either as a built-in YouTube mechanism, or more likely, as an add-on extension much like an adblocker on a web browser.

The objective of this study is to develop a model that could help to detect whether a video is a clickbait video or a non-clickbait video based of the title of the video. Even though there are already a lot of studies that uses the SVM, Na¨ıve-Bayes, and the LSTM model, we are aiming to improve the accuracy that is outputted by those 3 models The problem of classifying whether a content is clickbait or not has been a topic that has been researched since a long time ago, however, the problem of classifying clickbait YouTube videos in particular has only gained traction since 2018 [4].

A research from The British University of Egypt [9] used the SVM method and Logistic Regression that is trained on 2,495 texts which consists of 762 clickbait and 1,697 not clickbait posts and validated on 19,487 posts that consists of 14,774 clickbait posts and 4,713 non-clickbait posts. The result that they got is that SVM and Logistic Regression performed similarly with an accuracy of 74.5% on training and 79.4% on validation.

A research that is being published by the authors Deepika Varshney & Dinesh Kumar Vishwakarma [10] used a dataset of 987 videos which consists of 474 clickbait videos and 513 non-clickbait videos. In this research, they found that a lot of clickbait video titles have the words “Shocking”, “OMG”, “Sad News”, and “Bad News” on them. They got an accuracy of 98.89%.

According to the research conducted by BMS College of Engineering students [11], YouTube is becoming one of the major resources for consuming and sharing video content. That is why they do this research to detect titles that contain clickbait. They are using three methods for this research. Those methods are SVM, LSTM, and RF. From their research, they can get the precision of all three. For SVM, the accuracy is 96.76%, for LSTM, the accuracy is 93.79%, and for RF, the accuracy is 97.14%. They also analyzed the comments, views, likes, and dislikes to determine if the video is clickbait .

Another research compares 6 different models [12]. Those models are the K-Nearest Neighbor (KNN), Logistic Regression, Gaussian Na¨ıve Bayes, Extreme Gradient Boosting (XGBoost), Multi-Layer Perception, and Support Vector Machine (SVM). The models are tested on 2 different datasets. Those datasets are the BollyBAIT and MVD. BollyBAIT dataset is a dataset that contains 1000 video titles that contain 500 clickbait videos and another 500 non-clickbait videos which are all related to Bollywood. MVD is a dataset that contains the general multilingual YouTube title dataset. Where they got around 0.93 accuracy for the BollyBAIT dataset and 0.95 accuracy for the MVD dataset . A research conducted by Department of Computer Science and Engineering, University of Notre Dame [13], is using 9 methods, including AdaBoost (OVCP), LR, SVM, RF, MLP, VGG-16, ASONAM16, NGCT16, SPW18. From these 9 methods, the highest accuracy that they can get is 89.6% using the AdaBoost method.

2. Methodology

In order to do this project, we need to conduct model training. In order to support this model training we will conduct dataset selection, data processing, model selection, feature extraction, and ethical considerations beforehand. The training that will be conducted will be supervised learning using a data set that has been labeled.

2.1. Workflow

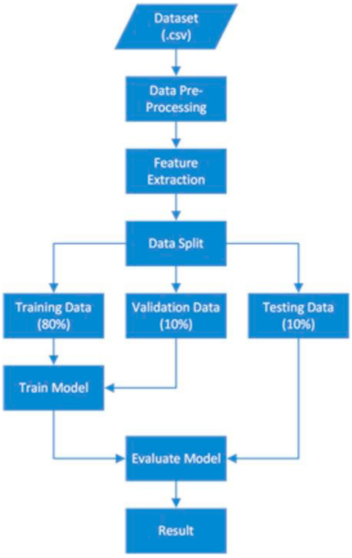

Fig. 1. Words Frequency in Dataset Using Word Cloud

The workflow on Fig. 1 represents how we do this research project. The first step that we do is looking for a compatible dataset for our experiments. After finding the dataset, we will do data preprocessing and feature extraction. Then we will split the dataset with a distribution ratio of 80% training data, 10% validation data, and 10% testing data. We will use the training data to train the model and validate it with the validation data. After that, the testing data is used to fine-tune and evaluate the model until it has been optimized to a satisfactory degree.

2.2. Dataset

Table 1. Dataset Example

Youtube Title	Label
15 Highly Important Questions About Adulthood, Answered By Michael Ian Black	1
250 Nuns Just Cycled All The Way From Kathmandu To New Delhi	1
How Much Would Chris Traeger Like You Based On Your Zodiac Sign	1
Australian comedians ”could have been shot” during APEC prank	0
Lycos launches screensaver to increase spammers’ bills	0
In Afghanistan, Soldiers Bridge 2 Stages of War	0
After Fleeing North Korea, an Artist Parodies Its Propaganda	0
Lessons (or Not) When a Start-Up Misses the Mark	0
Court Issues Order Against 3 Car-Warranty Calling Firms	0

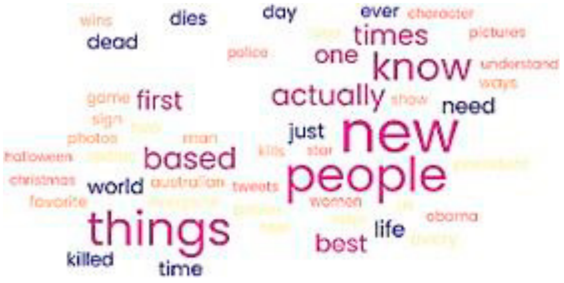

Fig. 2. Words Frequency in Dataset Using Word Cloud

The dataset we used is provided by Amit Chaudhary from GitHubGist where we show some of the dataset on Table 1 [14]. This dataset contained 31.987 titles and was divided into 25.590 titles for training the model, 6.397 titles used for validation of the model. From 6.937 validation set, we are going to divide it in half again, so the data split is not split into 80%, 10%, and 10% for training, validation, and test respectively. The data set is in the form of a csv file with two columns which are title and label. The title column contains the YouTube video titles, while the label column contains a classification of either '1' for containing clickbait or '0' for non-clickbait for determining whether a given video title is a clickbait or not. The most used words that are contained in the dataset are shown in Fig. 2. The dataset that is used for this research is publicly shared and can publicly be used for research purposes. There is also no bias that we can observe in the data selected for the dataset. The dataset consists of a lot of video titles from various content creators which are spread out popularity-wise. This project should also have no bias on creator following. Small and big creators will have the same judging based on how their titles are made. The word cloud from Fig. 2 shows the amount of a word contained in the whole dataset. It shows that the word "new" comes up a lot of times than other words in the dataset.

2.3. Data Preprocessing

From the dataset obtained, we will conduct data preprocessing in order to remove clutter from the dataset and make the classification process more accurate [15]. We will also remove stopwords as we found them to not have significant value in improving the outcome of the machine learning models that we are using [16]. However, we keptthe punctuations and capitalizations since we found them to be a significant part of clickbait titles [8, 17].

2.4. Feature Extraction

After all of the data preprocessing that have been conducted, we will do feature extraction with the goal of getting the representative features that will be used to train and test the model that we are experimenting with. We are using three feature extraction techniques and those techniques are Bag of Words, TF-IDF (Term Frequency-Inverse Document Frequency) and tokenization. In this feature extraction, we will take the preprocessed title and the label to help the model to determine whether the video title is a clickbait title or not.

2.5. Tokenization

Tokenization is used to separate each word on a sentence. Those separated words are what is called a token. The Word Tokenization technique provides a representation of the text used for training and validating the model. We are using word tokenization for the feature extraction techniques on LSTM. This is due to the sequential word tokenization that is being provided by the word tokenization technique which is compatible with RNN models such as LSTM [18].

2.6. Bag of Words

Bag of words is one of the most popular feature extraction techniques that is used on a machine learning model [19]. We are using the Bag of Words to do feature extraction for the SVM and Na¨ıve-Bayes model. This is because Bag of Words is more compatible to be used on traditional machine learning models than deep learning models. The Bag of Words feature extraction technique is very popular and commonly used due to its simplicity which doesn't require a complicated data preprocessing. But, due to its simplicity, it may cause high dimensionality and a loss of meaning in each word because of how the model ignores word order and information [20].

Table 2. Bag of Words Output Example.

About	Adulthood	All	Answered	Black	By	Cycled	Delhi	From	Highly	Ian	Important	Just	Kathmandu	Michael	New
1	1	0	1	1	1	0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	1	1	1	0	0	0	1	1	0	1

The way that Bag of Words work is by listing every unique word on the whole dataset, and then checking every word one by one. When Bag of Words finds a new word, it will create a new node in the matrix with the value 0. When the unique word is found, it will increase the value by one. So, if the unique word is now found, the value will change from 0 to 1. In Table 2, we are trying to simulate how Bag of Words works with the first 2 data (titles) from our dataset. The first row shows the existing words on the first sentence and the second row shows the existing word on the second sentence. This shows that the first sentence contains the words "About", "Adulthood", "Answered", "Black", "By", "Highly", "Ian", "Important, and "Michael" which represents the first sentence of the dataset which is "Highly Important About Adulthood Answered By Michael Ian Black".

2.7. TF-IDF

TF-IDF is also used to do feature extraction for the traditional machine learning models which we are using i.e., SVM and Na¨ıve-Bayes model. TF-IDF is used to determine the importance of a word within a sentence by weighting every word. This technique is usually used on text categorization. But TF-IDF feature extraction technique can't properly evaluate a word's weight when the word itself has come up for several times on the sentence [21].

Table 3. Bag of Words Output Example.

About	Adulthood	All	Answered	Black	By	Cycled	Delhi	From	Highly	Ian	Important	Just	Kathmandu	Michael	New
0.3	0.3	0	0.3	0.3	0.3	0	0	0	0.3	0.3	0.3	0	0	0.3	0
0	0	0.31	0	0	0	0.31	0.31	0.31	0	0	0	0.31	0.31	0	0.31

The way that TF-IDF works is by once again also list every unique word on the whole dataset and then checking every word one by one. But, instead of adding 1 every time TF-IDF found a same word, it will calculate the weight of the word which could range from 0 to 1. The higher the value, the more important the word to the dataset is. From Table 3, we are trying to simulate how TF-IDF works with the first 2 data from our dataset. The difference in the value from the first and the second data is due to the different amount of word on the data itself where the first sentence has 11 words, and the second sentence has 12 words. From the example on Table 3. We are using two sentences where the first row shows the existing words on the first sentence and the second row shows the existing word on the second sentence. This shows that the second sentence contains the words "All", "Cycled", "Delhi", "From", "Just", "Kathmandu", and "New" which represents the second sentence of the dataset which is "Just Cycled All From Kathmandu To New Delhi"

2.8. Model Selection

For the model selection here, we will use supervised learning where we use a dataset labeled as a clickbait video title or a non-clickbait video title. We will use 3 methods to classify each title and compare the accuracy of each method. Those methods are Naive-Bayes, LSTM (Long Short-Term Memory), and SVM (Support Vector Machine). LSTM, SVM, and Naive-Bayes methods can all be used to do classification. In this case, the classification we wanted to do is to classify whether a YouTube video title is a clickbait or not.

2.9. Naive-Bayes

By using Na¨ıve-Bayes, we are able to do text classification with high-speed and reasonable accuracy [22]. The Na¨ıve-Bayes algorithm is also considerably simpler than the others. Because the label on our dataset in this experiment, we are using the Bernoulli classifier for the Na¨ıve-Bayes model. The formula for using a Bernoulli Na¨ıve-Bayes model will be shown on equation 1 [23]:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

Where P(A—B) is the probability of A happening when B has already happened, P(B—A) is the probability of B happening when A has already happened, P(A) is the probability of A happening and P(B) is the probability of B happening [24].

2.10. LSTM(Long Short-Term Memory

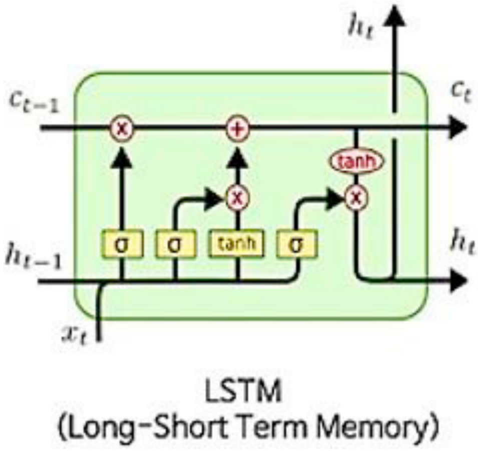

Fig. 3. LSTM Architecture

By using LSTM, the model is able to selectively remember and forget information so that the model will not be filled with insignificant information that is not relevant to the training of the model [25]. The visualization for LSTM is shown on Fig. 3. [26]:

While using LSTM, the features can be extracted with tokenizer. In the case of Naive-Bayes and SVM, the features can be extracted with bag-of-words or TF-IDF (Term Frequency-Inverse Document Frequency) approaches. We will train all the models with the same dataset and compare each method's accuracy, f1-score, precision, and recall. We use the tokenizer feature extraction method on LSTM because when we are doing text classification, the text can have a variable length. With tokenizer, we can ensure that the length of each title is consistent. When using Na¨ıve-Bayes and SVM, we are using Bag of Words and TF-IDF as the feature extraction method. We use TF-IDF and Bag of Words as the feature extraction method because TF-IDF and Bag of Words are both efficient to use on a large dataset. The reason we don’t use the same approach for LSTM is because TF-IDF and Bag of Words are both not suitable to use on a deep learning method such as LSTM.

2.11. SVM (Support Vector Machine)

SVM is commonly used and has shown great results in binary classification data, which is what we are going for in this project [27, 28, 29]. For SVM, we are going to use the kernel and gamma hyperparameter. We use linear kernel to judge whether the data that we are using are able to be separated linearly or not in order to determine accurately whether the video title is considered as a clickbait video title or not. We are also using gamma hyperparameter as a comparison with a low gamma value (0.1) because the size of the dataset is quite large. This is done in order to avoid overfitting the model.

3. Results and Discussion

The experiment is carried out by using Google Colab as a python IDE platform to run the experiment models. In this experiment, we are training three models broadly. Those models are SVM, Na¨ıve-Bayes, and LSTM. Here are the results that we got from the experiments that we have done:

Table 4. Result Summary

Algorithm	**Function**	**Feature Extractor**	**Metrics**			
			Accuracy	**Precision**	**Recall**	**F1-Score**
SVM	**Kernel**	BoW	98.32%	98.32%	98.32%	98.32%
		TF-IDF	**98.53%**	**98.53%**	**98.53%**	**98.53%**
	Gamma	BoW	98.12%	98.13%	98.13%	98.12%
		TF-IDF	97.53%	97.55%	97.53%	97.53%
Na¨ıve-Bayes	Bernoulli	BoW	98.12%	98.12%	98.12%	98.12%
		TF-IDF	98.12%	98.14%	98.12%	98.12%
LSTM	-	Tokenizer	97.94 %	97.94%	97.94%	97.94%

From the experiments we have done, we got the result represented by Table 4. We are using Bag of Words and TF-IDF feature extraction on SVM and Na¨ıve-Bayes model, while using the Tokenizer feature extraction on LSTM. The models are evaluated by using 4 metrics. Those metrics are “Accuracy”, “Precision”, “Recall”, and “f1-score”. On SVM, we are using the Kernel and Gamma hyperparameters while on Na¨ıve-Bayes, we are using the Bernoulli function. From Table 4, we can see that from the accuracy metric, LSTM got the lowest accuracy with 97.94%. Na¨ıve-Bayes got the same accuracy although using two different feature extraction techniques with 98.12%. Meanwhile, SVM got the highest accuracy with 98.53% using Kernel hyperparameter using TF-IDF feature extraction. All models are also able to achieve an optimal output based on the confusion matrix. We found that the minimum value of the True Positive is 1550 TP (97.05%) and 1548 TN (96.62%). Which means that our experiment models are able to achieve

Table 5. Comparison with Past Researches

Year	**Algorithm**	**Metrics**			
		Accuracy	**Precision**	**Recall**	**F1-Score**
2019	Linear Regression	-	79%	79%	79%
	SVM	-	78%	79%	79%
2020	RF	97.14%	95%	**100%**	98%
2021	J48	98.28%	98.3%	98.3%	-
	CPDM	95.3%	94.7%	95%	94.8%
2023	**SVM (Our Research)**	**98.53%**	**98.53%**	98.53%	**98.53%**

a minimum of 97.05% when predicting the “clickbait” labeled data and 96.62% when predicting the “non-clickbait” labeled data.

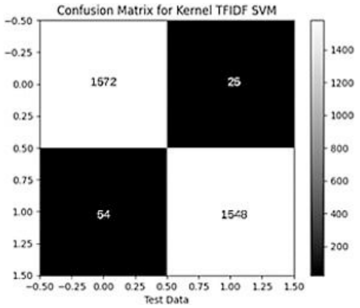

Fig. 4. Confusion Matrix

We found that all three models seem to perform excellently in all of the metrics that we tested with an average score of around 98% and no score below 97.53%. However, after comparing the results, we conclude that the SVM Kernel + TF-IDF model does slightly outperform other model variations scoring 98.53% across all metrics. This result shows that SVM, Na¨ıve-Bayes, and LSTM models are all capable of determining whether a YouTube video isclickbait or not based on the title that is used reliably. Our models also managed to outperform previous works modelsthat only managed to get 96.79% accuracy in SVM and 93.79% accuracy in LSTM models for YouTube clickbait title classification [11].

The best performing model is the SVM Kernel with TF-IDF feature extraction model that managed to score 98.53% on accuracy, precision, f1-score, and recall metrics. From the confusion matrix on Fig. 4, we can see that the model is able to correctly predict 1572 data as ‘1’ and able to correctly predict 1548 data as ‘0’.

We suspect that the reason the LSTM model performs underwhelmingly compared to SVM and Na¨ıve-Bayes is because the problem that we are currently researching is quite simple. The simplicity of the problem, which is only a binary classification might hurt the performance of LSTM and that would make the simpler algorithm such as SVM and Na¨ıve-Bayes have a better performance. The other reason is that the dataset that we are using is a nonsequential dataset where LSTM performs better when the dataset that is used is a sequential dataset. Meanwhile, the Na¨ıve-Bayes model’s performance is still quite strong, even when compared with some variations of the SVM model. This might be due to it only needing to perform simple binary classification on the dataset. Its lower score compared to the most optimal SVM model is majorly attributed to the simplicity of the model compared to SVM. The better performance of TF-IDF rather than Bag of Words might be due to the weighted features that are implemented in video titles which proved to be significant in detecting whether a video title is clickbait.

4. Conclusion

Clickbait classification is an important part in the broader fight against disinformation and increasing user satisfaction. This paper intends to add to the vast body of knowledge regarding clickbait, a simple and optimized way to detect clickbait videos on YouTube only from its title. In this work, we proved that we could reliably classify whether a YouTube video is clickbait or not with just the video title. We used 3 different models in this experiment, which are Na¨ıve-Bayes, SVM, and LSTM. We consistently achieved more than 97.53% accuracy on all of our models in multiple metrics, with the best performer, our kernel TF-IDF SVM model scoring 98.53% on accuracy, precision, recall, and f1-score.

In the future, we would like to see a classification model that considers more labels such as thumbnails, like- to-dislike ratio, comments, and other metadata in order to have a more complete picture of the video and increase accuracy. Aside from that, we’d also like a more nuanced dataset that gives a wider range of scores about the “clickbaitiness” of the video, in contrast to the current binary classification. Finally, some combined/hybrid machine learning models might be interesting to explore to increase accuracy results.

References

[1] Munger, K., 2020. All the news that's fit to click: The economics of clickbait media. Political Communication 37, 376–397. doi:10.1080/10584609.2019.1687626. publisher Copyright: © 2019, Copyright © 2019 Taylor Francis Group, LLC.

[2] Rony, M.M.U., Hassan, N., Yousuf, M., 2017. Diving deep into clickbaits: Who use them to what extents in which topics with what effects?, in: Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017, pp. 232–239

[3] Barwick, R., 2022. Advertisers spent $115 million on clickbait sites, report finds. URL: https://www.marketingbrew.com/stories/2022/07/18/advertisers-spent-usd115-million-on-clickbait-sites-report-finds

[4] Zannettou, S., Chatzis, S., Papadamou, K., Sirivianos, M., 2018. The good, the bad and the bait: Detecting and characterizing clickbait on youtube, in: 2018 IEEE Security and Privacy Workshops (SPW), IEEE. pp. 63–69.

[5] Grierson, S., 2022. Youtube vs. tiktok: Which is better for content creators? URL: https://www.backstage.com/magazine/article/youtube-vs-tiktok-which-is-better-75397/.

[6] Lu, Y., Pan, J., 2021. Capturing clicks: How the chinese government uses clickbait to compete for visibility. Political Communication 38, 23–54.

[7] Chauhan, V.K., Dahiya, K., Sharma, A., 2019. Problem formulations and solvers in linear svm: a review. Artificial Intelligence Review 52, 803–855.

[8] Kemm, R., . The linguistic and typological features of clickbait in youtube video titles. Social Communication 8, 66–80.

[9] Khater, S.R., Al-sahlee, O.H., Daoud, D.M., El-Seoud, M.S., 2018. Clickbait detection. Proceedings of the 7th International Conference on Software and Information Engineering doi:10.1145/3220267.3220287.

[10] Varshney, D., Vishwakarma, D.K., 2021. A unified approach for detection of clickbait videos on youtube using cognitive evidences. Applied Intelligence 51, 4214–4235.

[11] Vadde, N.R., Gupta, P., Mehta, P., Gupta, P., Vikranth, B., 2020. Analysis of youtube videos: Detecting click bait on youtube. International Journal of Scientific Engineering and Science 4, 15–17.

[12] Mowar, P., Jain, M., Goel, R., Vishwakarma, D.K., 2021. Clickbait in youtube prevention, detection and analysis of the bait using ensemble learning. arXiv preprint arXiv:2112.08611.

[13] Shang, L., Zhang, D.Y., Wang, M., Lai, S., Wang, D., 2019. Towards reliable online clickbait video detection: A content-agnostic approach. Knowledge-Based Systems 182, 104851.

[14] amitness, . clickbait.csv. URL: https://gist.github.com/amitness/0a2ddbcb61c34eab04bad5a17fd8c86b.

[15] Chakraborty, A., Paranjape, B., Kakarla, S., Ganguly, N., 2016. Stop clickbait: Detecting and preventing clickbaits in online news media. arXiv:1610.09786

[16] Silva, C., Ribeiro, B., 2003. The importance of stop word removal on recall values in text categorization, in: Proceedings of the International Joint Conference on Neural Networks, 2003., IEEE. pp. 1661–1666.

[17] Qu, J., Hißbach, A.M., Gollub, T., Potthast, M., 2018. Towards crowdsourcing clickbait labels for youtube videos., in: HCOMP (WIPDemo).

[18] Mielke, S.J., Alyafeai, Z., Salesky, E., Raffel, C., Dey, M., Gall e, M., Raja, A., Si, C., Lee, W.Y., Sagot, B., et al., 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. arXiv preprint arXiv:2112.10508.

[19] Zhang, Y., Jin, R., Zhou, Z.H., 2010. Understanding bag-of-words model: a statistical framework. International journal of machine learning and cybernetics 1, 43–52.

[20] Yan, D., Li, K., Gu, S., Yang, L., 2020. Network-based bag-of-words model for text classification. IEEE Access 8, 82641–82652.

[21] Gu, Y., Wang, Y., Huan, J., Sun, Y., Xu, S., 2020. An improved tfidf algorithm based on dual parallel adaptive computing model. International Journal of Embedded Systems 13, 18–27.

[22] Webb, G.I., Keogh, E., Miikkulainen, R., 2010. Na ıve bayes. Encyclopedia of machine learning 15, 713–714.

[23] Artur, M., 2021. Review the performance of the bernoulli na ıve bayes classifier in intrusion detection systems using recursive feature elimination with cross-validated selection of the best number of features. Procedia Computer Science 190, 564–570.

[24] Mishra, P., Biancolillo, A., Roger, J.M., Marini, F., Rutledge, D.N., 2020. New data preprocessing trends based on ensemble of multiple preprocessing techniques. TrAC Trends in Analytical Chemistry 132, 116045.

[25] Bahel, V., Pillai, S., Malhotra, M., 2020. A comparative study on various binary classification algorithms and their improved variant for optimal performance, in: 2020 IEEE Region 10 Symposium (TENSYMP), IEEE. pp. 495–498.

[26] Rahuljha, . Lstm gradients. URL: https://towardsdatascience.com/lstm-gradients-b3996e6a0296.

[27] Wickramasinghe, I., Kalutarage, H., 2021. Naive bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. Soft Computing 25, 2277–2293.

[28] Cui, C., He, M., Di, F., Lu, Y., Dai, Y., Lv, F., 2020. Research on power load forecasting method based on lstm model, in: 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), IEEE. pp. 1657–1660.

[29] Stanevski, N., Tsvetkov, D., 2005. Using support vector machine as a binary classifier, in: International Conference on Computer Systems and Technologies–CompSys Tech;