MACHINE LEARNING WITH DATA FRAME FOR CLASSIFICATION OF SPAM COMMENTS FROM YOUTUBE

B. Haritha Lakshmi[1], G. Vaishnavi[2], G. Sneha[2], G. Sushma[2]

[1,2]Department of Information Technology

[1,2]Malla Reddy Engineering College for Women (A), Maisammaguda, Medchal, Telangana.

ABSTRACT

YouTube, the world's largest video sharing site, was founded in 2005 and acquired by Google in 2006. YouTube has grown tremendously as a video content platform, with the recent shift in online content to video. At present, more than 400 hours of video are uploaded, and 4.5 million videos are watched every minute on YouTube. It is easy for users to watch and upload videos without any restrictions. This great accessibility has increased the number of personal media, and some of them have become online influencers. YouTube creators can monetize if they have more than 1,000 subscribers and 4,000 hours of watch time for the last 12 months. Accordingly, spam comments are being created to promote their channels or videos in popular videos. Some creators closed the comment function due to aggression such as political comments, abusive speech, or derogatory comments not related to their videos. YouTube has its own spam filtering system, though there are still spam comments that are not being caught.

Research on detecting spam content and users focus on various fields. Many studies focused on spam on websites (e.g., portal sites and blogs). As YouTube gains popularity as a video sharing platform, spammers target it with low quality content or promotions. Since spammers that harm the YouTube community are increasing, detecting them becomes an interesting source to research. Therefore, this project proposes a TF-IDF based machine learning technique to detect spam comments on YouTube, which have recently seen tremendous growth. YouTube is running its own spam blocking system but continues to fail to block them properly. Therefore, we examined related studies on YouTube spam comment screening and conducted classification experiments with supervised learning algorithm called multinomial naive Bayes.

Keywords: Spam comments, machine learning, YouTube.

1. INTRODUCTION

YouTube, the world's largest video sharing site, was founded in 2005 and acquired by Google in 2006. YouTube has grown tremendously as a video content platform, with the recent shift in online content to video. At present, more than 400 hours of video are uploaded, and 4.5 million videos are watched every minute on YouTube. It is easy for users to watch and upload videos without any restrictions. This great accessibility has increased the number of personal media, and some of them have become online influencers. YouTube creators can monetize if they have more than 1,000 subscribers and 4,000 hours of watch time for the last 12 months. Accordingly, spam comments are being created to promote their channels or videos in popular videos. Some creators closed the comment function due to aggression such as political comments, abusive speech, or derogatory comments not related to their videos. YouTube has its own spam filtering system, though there are still spam comments that are not being caught.

2. LITERATURE SURVEY

Oh et al. examined related studies on YouTube spam comment screening and conducted classification experiments with six different machine learning techniques (Decision tree, Logistic regression, Bernoulli Naïve Bayes, Random Forest, Support vector machine with linear kernel, Support vector machine with Gaussian kernel) and two ensemble models (Ensemble with hard voting, Ensemble with soft voting) combining these techniques in the comment data from popular music videos - Psy, Katy Perry, LMFAO, Eminem and Shakira.

Tashtoush et al. classified these comments using different algorithms such as Decision Tree (DT), Support Vector Machine (SVM), Naive Bayes (NB), Random Forest, and k-Nearest Neighbor (k-NN).

Abdullah et al. conducted a comparative study of the common filtering techniques used for YouTube comment spam. The study deployed datasets extracted from YouTube using its Data API. According to the obtained results, high filtering accuracy (more than 98%) can be achieved with low-complexity algorithms, implying the possibility of developing a suitable browser extension to alleviate comment spam on YouTube in future.

Das et al. taken YouTube comment datasets of five famous singers and detecting Spam comments using some Artificial Neural Network based Classifiers and some Normal Classifiers. The proposed technique compared the derived results of the classifiers and suggests the best classifiers for detecting Spam comments.

Ezpeleta et al. focused on mood analysis, among all content-based analysis techniques. This work demonstrated that using this technique social spam filtering results are improved. First, the best spam filtering classifiers are identified using a labeled dataset consisting of Youtube comments, including spam. Then, a new dataset is created adding the mood feature to each comment, and the best classifiers are applied to it.

Aziz et al. developed a YouTube detection framework by using Support Vector Machine (SVM) and K-Nearest Neighbor (k-NN). There are five (5) phases involved in this research such as Data Collection, Pre-processing, Feature Selection, Classification and Detection. The experiments are done by using Weka and RapidMiner. The accuracy result of SVM and KNN by using both machine learning tools show good accuracy result. Others solution to avoid spam attack is trying not to click the link on comments to avoid any problems.

Kavitha et al. categorized the user comments posted on YouTube video sharing website based on their relevance to the video content given by the description associated with the video posted. Comments are analysed for polarity and are further segregated as positive or negative. A comparative analysis of classifier using the Bag of Words and Association List approaches is presented.

Khodake et al. evaluated several top-performance classification techniques for detecting and analyzing spam comments. The statistical analysis of results indicates that, with 99.9% of confidence level, decision trees, logistic regression, Bernoulli Naive Bayes, random forests, linear and Gaussian SVMs are statistically equivalent in maximum rate. Therefore, it is very important to find a way to detect these comments on videos and report them before they are viewed by innocent users.

Jamalludin et al. classified comments using the Support Vector Machine (SVM) algorithm. While the method, the gain ratio is used by the author for the feature selection process stage to help the performance of the Support Vector Machine (SVM) algorithm where the features used are from 100% features to 5% features so that they get different Accuracy, Precision, Recall values. From the results of research conducted by the author, the best precision results are at 50% features, while for accuracy and recall are at 5% features.

Chetty et al. proposed a deep learning-based spam detection model. This model is a combination of the Word Embedding technique and Neural Network algorithm. Word Embedding allows a distributed representation of words in the feature space where word's meaning and word analogy can be represented. Deep neural network is used to learn features of text documents represented in the embedding space and use these features to classify text documents. This model architecture is expected to be able to effectively detect spams in various types of text documents as well as in large document corpus.

Samsudin et al. proposed the YouTube detection framework, examined, and validate each of the phases by using two types of data mining tool. The features are constructed from analysis by using data collected from YouTube Spam dataset by using Naïve Bayes and Logistic Regression and tested in two different data mining tools which is Weka and Rapid Miner. From the analysis, thirteen (13) features that had been tested on Weka and RapidMiner shows high accuracy, hence is being used throughout the experiment in this research. Result of Naïve Bayes and Logistic Regression run in Weka is slightly higher than RapidMiner. In addition, result of Naïve Bayes is higher than Logistic Regression with 87.21% and 85.29% respectively in Weka. While in RapidMiner there is slightly different of accuracy between Naïve Bayes and Logistic Regression 80.41% and 80.88%. But precision of Naïve Bayes is higher than Logistic Regression.

3. PROPOSED METHOD

Therefore, this project proposes a TF-IDF based machine learning technique to detect spam comments on YouTube, which have recently seen tremendous growth. YouTube is running its own spam blocking system but continues to fail to block them properly. Therefore, we examined related studies on YouTube spam comment screening and conducted classification experiments with supervised learning algorithm called SVM.

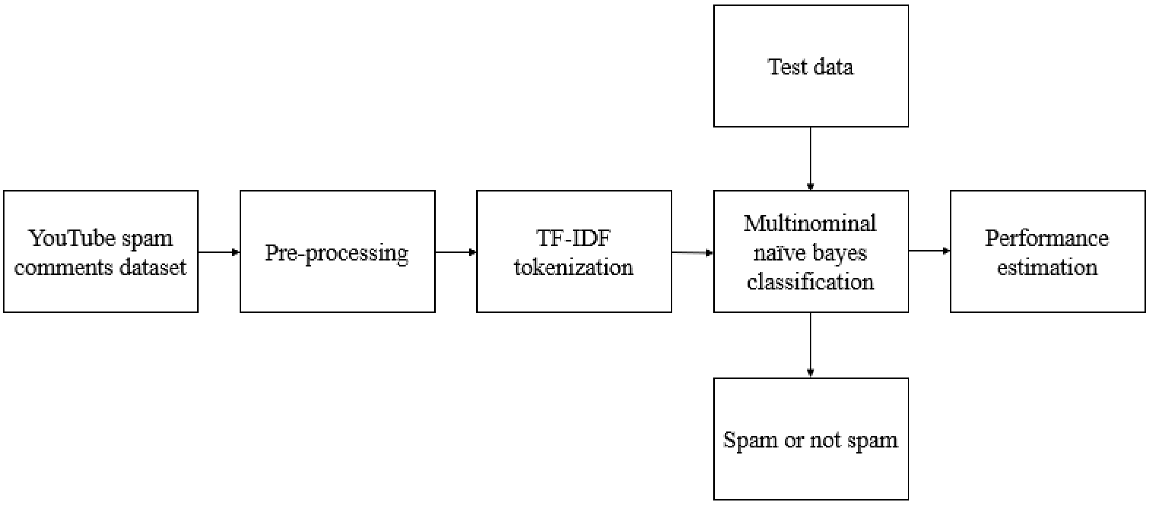


Fig. .1: Block diagram of proposed system.

3.1 Dataset Description

5- Columns: Comment-ID, Author, Date, Content, Class.

351-Rows

3.2 Pre-processing

Data Pre-processing in Machine learning

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task.

Why do we need Data Pre-processing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

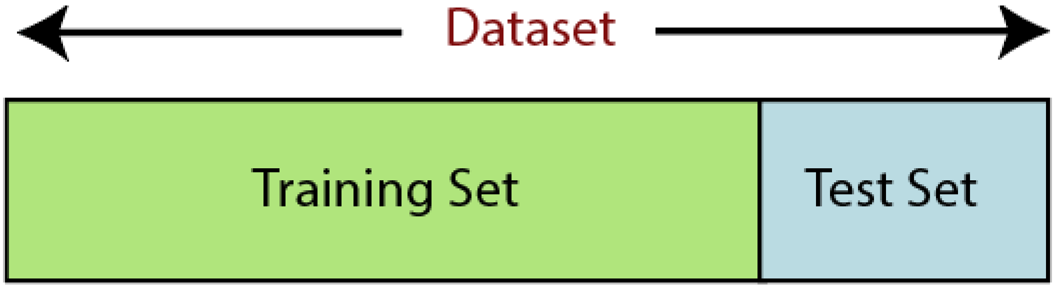
- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set
- Feature scaling

3.2.1 Splitting the Dataset into the Training set and Test set

In machine learning data pre-processing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model.

Supposeif we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:



Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

3.3 TF-IDF

TF-IDF which stands for Term Frequency – Inverse Document Frequency. It is one of the most important techniques used for information retrieval to represent how important a specific word or phrase is to a given document. Let's take an example, we have a string or Bag of Words (BOW) and we have to extract information from it, then we can use this approach.

The tf-idf value increases in proportion to the number of times a word appears in the document but is often offset by the frequency of the word in the corpus, which helps to adjust with respect to the fact that some words appear more frequently in general. TF-IDF use two statistical methods, first is Term Frequency and the other is Inverse Document Frequency. Term frequency refers to the total number of times a given term t appears in the document doc against (per) the total number of all words in the document and The inverse document frequency measure of how much information the word provides. It measures the weight of a given word in the entire document. IDF show how common or rare a given word is across all documents. TF-IDF can be computed as tf * idf

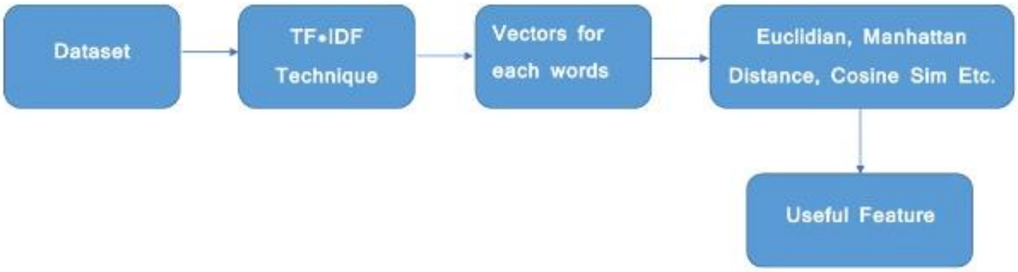


Fig. 2: TF-IDF block diagram.

TF-IDF do not convert directly raw data into useful features. Firstly, it converts raw strings or dataset into vectors and each word has its own vector. Then we'll use a particular technique for retrieving the feature like Cosine Similarity which works on vectors, etc.

Terminology

t — term (word)

d — document (set of words)

N — count of corpus

corpus — the total document set

Step 1: Term Frequency (TF): Suppose we have a set of English text documents and wish to rank which document is most relevant to the query, "Data Science is awesome!" A simple way to start out is by eliminating documents that do not contain all three words "Data" is", "Science", and "awesome", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document; the number of times a term occurs in a document is called its term frequency. The weight of a term that occurs in a document is simply proportional to the term frequency.

$$tf(t,d) \ = \ count\ of\ t\ in\ d\ /\ number\ of\ words\ in\ d$$

Step 2: Document Frequency: This measures the importance of document in whole set of corpora, this is very similar to TF. The only difference is that TF is frequency counter for a term t in document d, whereas DF is the count of occurrences of term t in the document set N. In other words, DF is the number of documents in which the word is present. We consider one occurrence if the term consists in the document at least once, we do not need to know the number of times the term is present.

$$df(t) \ = \ occurrence\ of\ t\ in\ documents$$

Step 3: Inverse Document Frequency (IDF): While computing TF, all terms are considered equally important. However, it is known that certain terms, such as “is”, “of”, and “that”, may appear a lot of times but have little importance. Thus, we need to weigh down the frequent terms while scale up the rare ones, by computing IDF, an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely. The IDF is the inverse of the document frequency which measures the informativeness of term t. When we calculate IDF, it will be very low for the most occurring words such as stop words (because stop words such as “is” is present in almost all of the documents, and N/df will give a very low value to that word). This finally gives what we want, a relative weightage.

$$idf(t) = N/df$$

Now there are few other problems with the IDF, in case of a large corpus, say 100,000,000, the IDF value explodes, to avoid the effect we take the log of idf . During the query time, when a word which is not in vocab occurs, the df will be 0. As we cannot divide by 0, we smoothen the value by adding 1 to the denominator.

$$idf(t) = log(N/(df + 1))$$

The TF-IDF now is at the right measure to evaluate how important a word is to a document in a collection or corpus. Here are many different variations of TF-IDF but for now let us concentrate on this basic version.

$$tf - idf(t,d) = tf(t,d) * log(N/(df + 1))$$

Step 4: Implementing TF-IDF: To make TF-IDF from scratch in python, let’s imagine those two sentences from different document:

first sentence: “Data Science is the sexiest job of the 21st century”.

second sentence: “machine learning is the key for data science”.

3.4 Support Vector Machine Algorithm (SVM)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

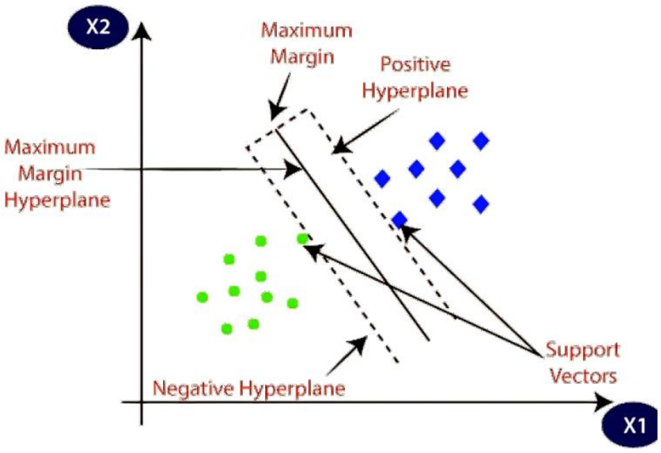


Fig. 3: Support vector machine.

3.5 Advantages of proposed system

- SVM works relatively well when there is a clear margin of separation between classes.
- SVM is more effective in high dimensional spaces.
- SVM is effective in cases where the number of dimensions is greater than the number of samples.
- SVM is relatively memory efficient.

4. RESULTS AND DISCUSSION

4.1 Module implementation

- Import dataset files
- Splitting data
- Apply TFIDF vectorizer
- Training
- Multinominal naive bayes model
- Predict test data
- Performance metrics

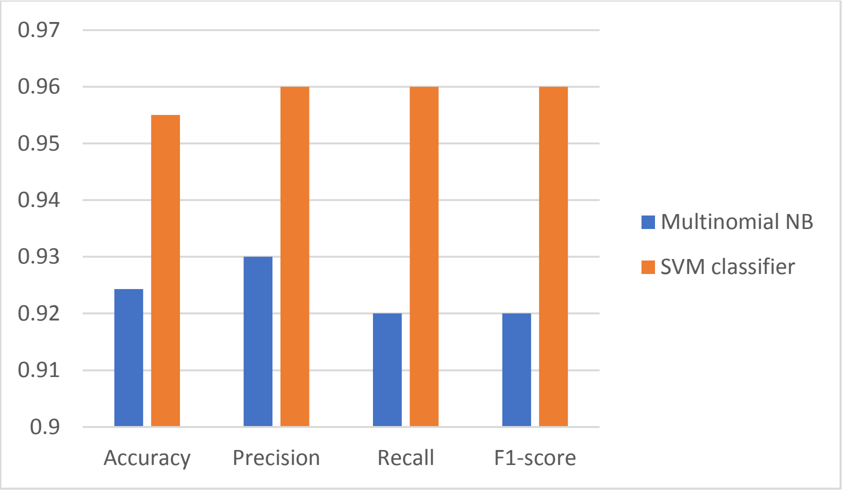


Fig. 4: Performance comparison of quality metrics obtained using existing and proposed YouTube spam detection classifier.

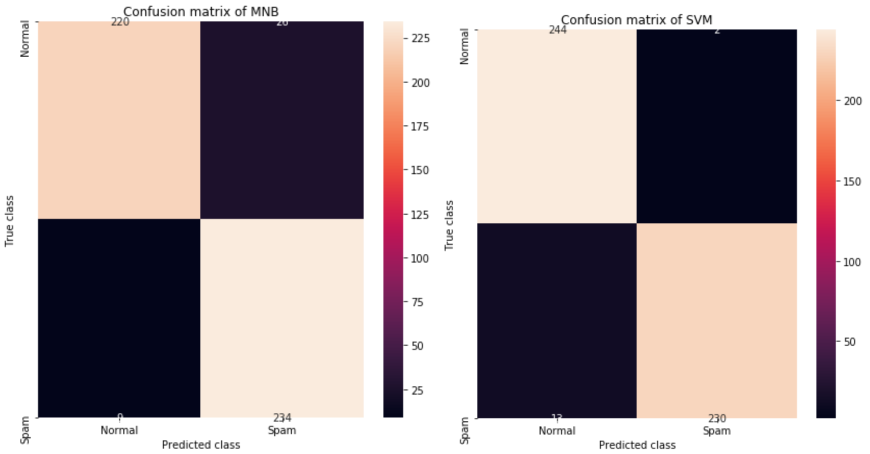
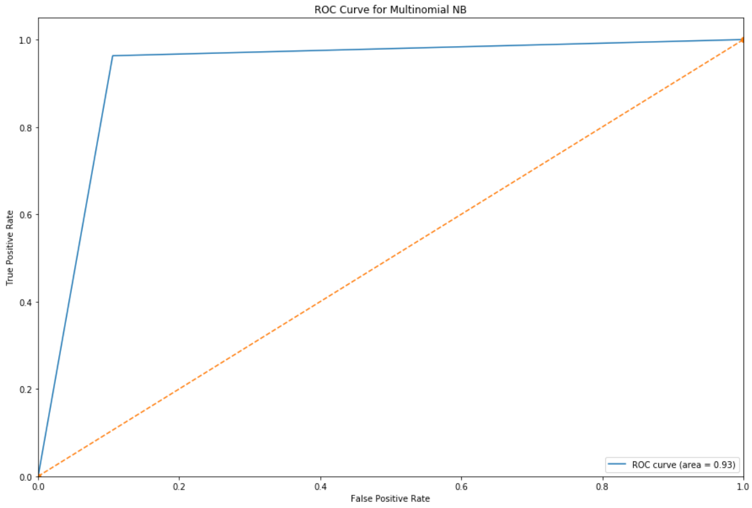
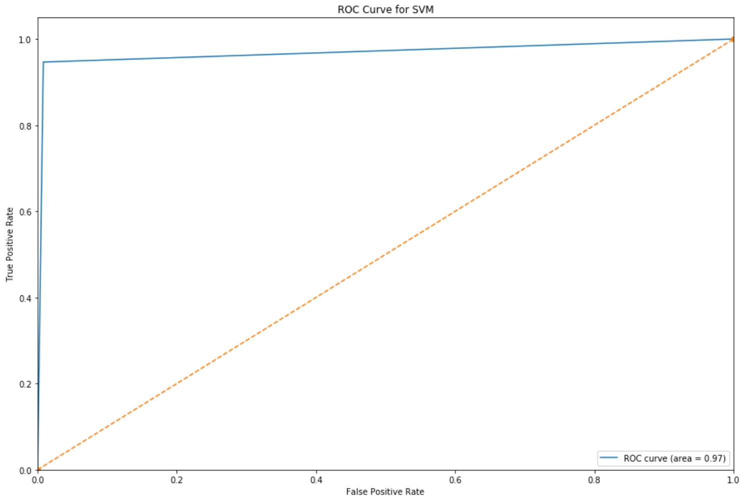


Fig. 5: Obtained confusion matrices for existing and proposed spam classifier system.





5. CONCLUSION

In this research, the development of a spam comment detection framework by using machine learning techniques has been done. It is important to improve security since the Internet nowadays that indication the security issue. There are many studies aimed to reduce attack and to protect user privacy but lacking in applying the techniques for social media. This paper also wants to contribute by examining the suitable features based on the real comment from social media site for developing spam comment detection framework.

Future Scope

Future scope of this work is detection of intrusive comments or spam on the video-sharing website - Youtube. We describe spam comments as those which have a promotional intent or those who deem to be contextually irrelevant for a given video. The prospects of monetisation through advertising on popular social media channels over the years has attracted an increasingly larger number of users. This has in turn led to to the growth of malicious users who have begun to develop automated bots, capable of large-scale orchestrated deployment of spam messages across multiple channels simultaneously. The presence of these comments significantly hurts the reputation of a channel and the experience of normal users. Youtube themselves have tackled this issue with very limited methods which revolve around blocking comments that contain links. Such methods have proven to be extremely ineffective as Spammers have found ways to bypass such heuristics. Standard machine learning classification algorithms have proven to be somewhat effective but there is still room for better accuracy with new approaches.

REFERENCES

[1] S. Aiyar and N. P. Shetty, ‘‘N-gram assisted Youtube spam comment detection,’’ Proc. Comput. Sci., vol. 132, pp. 174–182, Jan. 2018, doi: 10.1016/j.procs.2018.05.181.

[2] A. Kantchelian, J. Ma, L. Huang, S. Afroz, A. Joseph, and J. D. Tygar, ‘‘Robust detection of comment spam using entropy rate,’’ in Proc. 5th ACM Workshop Secur. Artif. Intell. (AISec), 2012, pp. 59–70, doi: 10.1145/2381896.2381907.

[3] H. Oh, "A YouTube Spam Comments Detection Scheme Using Cascaded Ensemble Machine Learning Model," in IEEE Access, vol. 9, pp. 144121-144128, 2021, doi: 10.1109/ACCESS.2021.3121508.

[4] Y. Tashtoush, A. Magableh, O. Darwish, L. Smadi, O. Alomari and A. ALghazoo, "Detecting Arabic YouTube Spam Using Data Mining Techniques," 2022 10th International Symposium on Digital Forensics and Security (ISDFS), 2022, pp. 1-5, doi: 10.1109/ISDFS55398.2022.9800840.

[5] A. O. Abdullah, M. A. Ali, M. Karabatak and A. Sengur, "A comparative analysis of common YouTube comment spam filtering techniques," 2018 6th International Symposium on Digital Forensic and Security (ISDFS), 2018, pp. 1-5, doi: 10.1109/ISDFS.2018.8355315.

[6] Das, R.K., Dash, S.S., Das, K., Panda, M. (2020). Detection of Spam in YouTube Comments Using Different Classifiers. In: Pati, B., Panigrahi, C., Buyya, R., Li, KC. (eds) Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, vol 1082. Springer, Singapore. https://doi.org/10.1007/978-981-15-1081-6_17

[7] Ezpeleta, E., Iturbe, M., Garitano, I., de Mendizabal, I.V., Zurutuza, U. (2018). A Mood Analysis on Youtube Comments and a Method for Improved Social Spam Detection. In: , et al. Hybrid Artificial Intelligent Systems. HAIS 2018. Lecture Notes in Computer Science (), vol 10870. Springer, Cham. https://doi.org/10.1007/978-3-319-92639-1_43

[8] Aziz, A. & Mohd Foozy, Cik Feresa & Shamala, Palaniappan & Suradi, Zurinah. (2018). Youtube spam comment detection using support vector machine and K–nearest neighbor. Indonesian Journal of Electrical Engineering and Computer Science. 12. 607-611. 10.11591/ijeecs. v12.i2.pp607-611.

[9] K.M. Kavitha, Asha Shetty, Bryan Abreo, Adline D’Souza, Akarsha Kondana, “Analysis and Classification of User Comments on YouTube Videos”, Procedia Computer Science, Volume 177, 2020, Pages 593-598, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.10.084.

[10] Khodake, Swati, et al. "Analysis and Detection of Spam Comments on Social Networking Platforms like YouTube using Machine Learning."

[11] M. D. Jamalludin, Muljono, G. Fajar Shidik, A. Zainul Fanani, Purwanto and F. Al Zami, "Implementation of Feature Selection Using Gain Ratio Towards Improved Accuracy of Support Vector Machine (SVM) on Youtube Comment Classification," 2021 International Seminar on Application for Technology of Information and Communication (iSemantic), 2021, pp. 28-31, doi: 10.1109/iSemantic52711.2021.9573191.

[12] G. Chetty, H. Bui and M. White, "Deep Learning Based Spam Detection System," 2019 International Conference on Machine Learning and Data Engineering (iCMLDE), 2019, pp. 91-96, doi: 10.1109/iCMLDE49015.2019.00027.

[13] N. M. Samsudin, C. F. B. Mohd Foozy, N. Alias, P. Shamala, N. F. Othman and W. I. S. Wan Din, "Youtube spam detection framework using Naïve Bayes and logistic regression", Indonesian J. Electr. Eng. Comput. Sci., vol. 14, no. 3, pp. 1508, Jun. 2019.