# YouTube Comments Spam Detection

Harish Kugur Sreekanta Babu, *Univeristy of Florida*

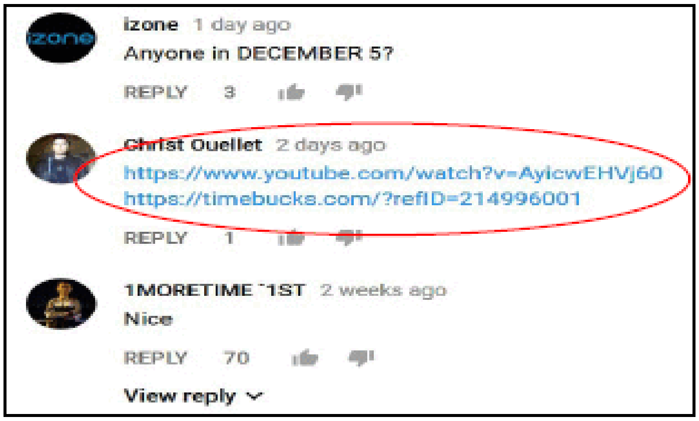
***Abstract*— This project proposes the method for distinguishing the comments on YouTube as Spam and Ham i.e. comments those are spam and not-spam. This approach would aid in constructing an algorithm for blocking/removing inadmissible and undesirable comments from the view. The data set of 5 different YouTube Videos of 1956 comments available from UCI Machine Learning Repository are analyzed for the proposed work. Data set has 5 CSV files that features video ID, comment ID, Author name, Date on the comment posted, Content of the comment and the tag. Primarily, data set will be pre-processed and redundant data will be excluded. In subsequent phases, keywords (excluding common English words) are extracted using bag-of-words method and the frequency of usage is computed. Similarly, in the subsequent phases, essential features are derived by adding weight values to the data using tf-idf feature extraction technique to form the distinct components of data used for further evaluation. On discrete data, Machine Learning Modelling will be applied. Logistic Regression, Random Forest Classifier, Support Vector Machine, Multi-layer Perceptron models are exercised on the refined training data to predict the whether a comment is a Spam or Ham based on the patterns recognized. Lastly, testing data is used to validate the trained models. Out of the models employed, Random Forest Classifier produced best accuracy of 94.13%. This mode of approach would form the basis for further spam detection on various social media platforms.**

***Index Terms*—YouTube Comments, Spam, Ham, Data Processing techniques, Supervised Machine Learning Methods, Classification, Regression, Python Libraries, SciKit, TF-IDF, Logistic Regression, Support Vector Machine, Random Forest Classifier, Multi-layer perceptron,**

## I. Introduction

YouTube, the World's Most Viewed Video Sharing Platform. It presents the user base all categories of Information, News, Facts, Music, Live Telecasts of Sports, Concerts and even millions of channels that can exhibit their content.

In the information age, there has been tremendous amount of knowledge shared through various ways. Even the sources of the information have been increased by ample amount. Voluminous advices, suggestions, entertainment is available at ease for the consumers. According to a market trend analysis of social media platforms, it is known that YouTube has more than 2 billion users and 500 hours of video is uploaded for every 60 seconds 43,200,000 hours of video on every day [1-3]. This statistics also collaterally indicates the earning of millions of dollars. In this view, much of the YouTubers request their users for Likes, Shares, Comments and Subscribes as it would earn them larger user base and hence bountiful feast of earnings. Along with the pros, there's always associated downsides and here, that is Comments – more specifically Spams as shown in Fig 1. Discussion and analysis of various happenings in the world and related matter in the comment section is welcoming as it is also plays role in educating the users. But Online Inclivity, Spamming are one of the contemporary issues that has entangled the world [4]. Even though, YouTube has spam reporting mechanism, ascertaining comments as spam and control of it has not been fully successful. The proposed work would make the nascent steps into detecting Spams that should be eliminated and this should form the footing ground for further detection of Online Inclivity and removal of them.



Izone 1 day ago

Anyone in DECEMBER 5?

REPLY 3

Christ Ouellet 2 days ago

https://www.youtube.com/watch?v=AyicwEHVj60
https://timebucks.com/?refID=214996001

REPLY 1

1MORETIME `1ST 2 weeks ago

Nice

REPLY 70

View reply ⌄

Fig. 1. Spam comments representation on YouTube

## II. Motivation

YouTube is the most viewed social platform since its commencement in 2005. Gradually, spam was spread to YouTube which was until then affected the Emails. The spam content on YouTube is analogous to the profit created by marketing and propagation of undesired and malicious content. Due to multiple complaints from the users, YouTube build a technique to identify malicious links and virus websites through ASCII art in 2013 but still this method wasn't sufficient enough to counteract the spam issues. A channel that had highest subscribers then even had disabled the comment section to avoid the spam and malicious content from being diffused into the public domain according to article published by The Guardian [23]. Recently, pedophile ring was observed on YouTube comments and hence the company announced that it would disable comments for the videos featuring children as the detection and handling of spam is not efficiently in place yet. This is not a good sign for the Social Media giant as Channel Owners develop public relations through comments section and therefore disabling it wouldn't be the solution for this. Hence all these persistent issues call for the proposed work of detecting the spam comments efficiently by collecting the significant information from the comments and discarding the unwanted content and then applying Machine Learning Models on the Training and Testing Data to evaluate the efficiency of the algorithm.

## III. Related Work

A spam detection methodology constructed by Alex Kantchelian et al [18], identified redundant and futile blogs to filter and to provide productive and significant blogs for user reading. They suggested to extend the technique to identify the spams. This model was fruitless for YouTube Spam detection as the comments on YouTube are usually of smaller lengths and hence the technique created for blogs spam detection was not applicable here.

A method is discussed in [19], [20] to distinguish spam and ham in emails and social media sites. But this technique fails in identifying spam in YouTube, because in other social media sites and emails, generally spams are generated by Bots. On the other hand, spam in YouTube is usually created by actual users to self-promote and market their brands or propagating malicious links and websites to hack the accounts. So method proposed cannot be applied as spam content might look similar to the authentic comments, for example: Links of the most famous or most watched videos of an artist/music album will be shared in other related videos too and such comments cannot be tagged as Spam Comment.

Another technique proposed by M. McCord in [21] used twitter data set to test their algorithm and have used machine learning algorithms to classify comments and users as spam and spammers respectively. However the technique backslides in identifying the spams and spammers on YouTube as the algorithm is designed to work on suggested user and content-based data where in, latest 100 tweets of each user and their followers are examined to classify as spam or not. Equivalently, content-based data analysis cannot be applied on YouTube as it is mainly a video sharing application and not a microblogging application like Twitter.

Additionally, an algorithm designed to find the spam comments on Ted videos by comparing the affinity of content on videos to the comments on each video. Based on amount of divergence, comments were classified as spam or not. Practically, this cannot be applied on YouTube as the content on each video and on each channel is different. So capturing the content and comparing the affinity is unattainable [22].

## IV. Description

### A. Dataset

The proposed work is applied on the dataset that consists of comments from five of the most watched YouTube videos by 2015. The data is captured through YouTube Data API link and available for public use [5]. The dataset has 5 csv files that corresponds to five most watched videos respectively, which totally has 1956 comments. The dataset has following details:

- COMMENT_ID : Unique id for each comment
- AUTHOR : Name of the user who has commented
- DATE : Date on which the comment was published with time stamp
- CONTENT : Content of the comment
- CLASS: 1 or 0

Following are the data files used in the project:

1. 'Youtube01Psy.csv'
2. 'Youtube02-KatyPerry.csv'
3. 'Youtube03-LMFAO.csv'
4. 'Youtube04-Eminem.csv',
5. 'Youtube05-Shakira.csv'.

All the comments in these files are labelled as spam or ham (legitimate), where class value 1 corresponds to spam and 0 corresponds to ham comments. Fig 2, shows the common type of spam comments used on the internet.

Fig. 2. Common Spam comments on Internet

### B. Cleaning the Data

Data cleaning is the phenomenon of finding wrong - formatting, construction, spellings, unrequired characters, spaces and etc. As data is the core component of data analysis, cleaning the data to format it to the requirement is imperative. After reading the data from all the five 5 files, it is added to the data files array

In the proposed work, data cleaning is done in two stages:

i. Removing insignificant columns: Comment_ID, Author and Date Columns are removed as they do not add much value for spam comment analysis. Now, the data files array contains only Content and Class columns

ii. Processing the Content Column: Content i.e. comments on YouTube might contain lot of punctuations, special characters, numbers and other superfluous content. In this stage, only the characters ranging from A-Z are retained and all the characters in the comments are converted to lowercase.

### C. *Splitting and Sampling data*

The processed data now should be split into training and testing data sets. As the name suggests, training data set is used for training the Machine Learning Models in further stages. To test the accuracy of model training, test data is used. An important aspect of this stage is – whole process should be done randomly as it yields in better training the models as it produces less biased data sets. To accomplish to random splitting and Sampling of data, python library SciKit – splitter function is used.

SciKit library offers various splitter functions such as KFold, StratifiedKFold, StratifiedShuffleSplit, train_test_split etc.

- KFold – Splits the data into 'k' folds without shuffling by default

- StratifiedKFold – Splits the data into 'k' folds similarly like KFold, along with that this function makes sure that certain percentage of input samples are present in each fold by shuffling the data once before splitting. In the process, one of the fold is picked as test data and remaining as training data

- StratifiedShuffleSplit – This function first shuffles the data and splits into n splits. One of the part is picked as a test data and again the remaining data is shuffled and splits into n-1 splits. Because of multiple shuffles, test data might have overlaps

- Train_test_split: This function works in a simple manner by splitting the data randomly into training and test sets. It provides few parameters such as test_size, train_size, random_state, shuffle and etc.

In this project, **train_test_split** function is used to split the 80% of the data as training set and remaining 20% as test set by specifying test_size = 0.2 and random state=57 that constitutes 1563 comments as training and 393 comments as testing samples. Random State parameter facilitates random splitting of data as training and testing set. X parameter is the Processed Content column and Y parameter is the CLASS column.

### D. *Feature Extraction*

On completion of Cleaning and splitting the data into training and testing samples, it entails the process of Feature Extraction. Data set now contains the comments and its corresponding class value suggesting whether it is a Spam or a ham. Comments are a string of words and therefore *bag-of-words* model can be correlated on the data to extract the all words from all the sentences and then finding its size in the bag.

For an example, Let following be the 5 comments in the data set:

i. "hi how are you"
ii. "hi who are you"
iii. "i like this song"
iv. "hi do you like the song"
v. "hi click on the link"

Applying *Bag-of-Words* technique on this, all the words from all the comments are extracted,
**['hi', 'how', 'are', 'you', 'who', 'i', 'like', 'this', 'song', 'do', 'the', 'click', 'on', 'link']**

On closer observation, it can be noticed that all the words are treated as individual entities and duplicate terms are not considered for the bag. This data cannot be fed into the models directly as the machine does not understand the context of these words. Next task is to convert the extracted words into the binary language.

For conversion to binary language, let us find the multiplicity of the words i.e.

TABLE I
COUNT VALUES OF EACH WORD

| Terms | Occurrence Count |
|---|---|
| hi | 4 |
| how | 1 |
| are | 2 |
| you | 2 |
| who | 1 |
| i | 1 |
| like | 2 |
| this | 1 |
| song | 2 |
| do | 1 |
| the | 2 |
| click | 1 |
| on | 1 |
| link | 1 |

Subsequently, vectors can be formed using the extracted words and the data set will be created as the following:

i. “hi how are you” = [1,1,1,1,0,0,0,0,0,0,0,0,0,0]
ii. “hi who are you” = [1,0,1,1,1,0,0,0,0,0,0,0,0,0]
iii. “i like this song” = [0,0,0,0,0,1,1,1,1,0,0,0,0,0]
iv. “hi do you like the song” = [1,0,0,1,0,0,1,0,1,1,1,0,0,0]
v. “hi click on the link” = [1,0,0,0,0,0,0,0,0,0,1,1,1,1]

The above explained process is implemented using ***SciKit-CountVectorizer***. CountVectorizer function invokes both ‘tokenization’ i.e. extracting words and ‘counting’. It also creates vectors that assigns the binary value to each word on the row basis. In the proposed work – each word are treated as Unigram.

The data created using results in tokens of words. But this might reflect the actual context of spam comments on YouTube, because certain words in English like – ‘a’, ‘the’, ‘and’ etc. are used profusely in the sentences. Inclusion of such words in the comments might not actually indicate any resemblance to spam comments. There such ‘stop words’ are removed by passing the parameter – “*stop_words: english”*. *SciKit* already has a list of unhelpful words and removes it from the bag of words.

Now, we have list of words and its occurrence count in the comments. As shown in the above example, ‘hi’ has a total count of 4 and ‘click’, ‘link’ – have total count of 1. In the context of identifying the spam comments – a term that has the highest frequency cannot be considered as the indicative. Apparently, even in the above example, ‘hi’ does not have much correspondence towards being the spam word. Contrarily, words like ‘click’, ‘link’ may have affinity towards being the spam words even though their frequency is just one. If this data is fed to the model for training, then the words which have less meaningful info might obscure the actual spam words and hence necessitates the process of re-weighting the terms.

To reweight the terms based on the context of the sentences, SciKit ‘*Tf-idf term weighting*’ function is employed on the vectorized data. “Tf means term frequency and idf means inverse document frequency”

$$\text{tf-idf(t,d)} = \text{tf(t,d)} \times \text{idf(t)} \quad (1)$$

$$\text{idf}(t) = \log \frac{1+n}{1+\text{df}(t)} + 1 \quad (2)$$

$$v_{norm} = \frac{v}{||v||_2} = \frac{v}{\sqrt{{v_1}^2 + {v_2}^2 + \cdots + {v_n}^2}} \quad (3)$$

The above equations (1), (2) and (3) are captured from https://scikit-learn.org/ [7]

Equation (1) explains that ***tf-idf*** is a product of tf and idf.

Equation (2), represents the way of calculating inverse document frequency of a term (t), where

n - the total number of comments in the data set
df (t) - the total number of comments that contain a term – t in the whole data set.

If a term frequency is high in the data set then its idf value will be low and vice-versa. In this way, IDF ascertains the weight as less frequently used words in the data set might add value in detecting spam words.

Equation (3) – vectors resulted in equation (2) i.e. weights assigned in the IDF equation is normalized using Euclidean norm in equation (3). That is, process of transforming the data elementarily means to “center the data set” – subtract the mean and divide the outcome by the standard deviation to prepare the data set that has zero mean and unit standard deviation.

TABLE II
TF-IDF VALUES FOR THE TERMS [ONLY FOR REPRESNTATIVE PURPOSE]

| Terms | Tf-idf |
|---|---|
| hi | 0.203 |
| you | 0.452 |
| click | 1.891 |
| link | 2.322 |

As shown in the Table II, the TF-IDF values for the less frequently used terms are high and frequently used terms have relatively lower values. Note: These values are only for representative purpose and does not indicate actual values.

Similarly in the project, TF-IDF reassigns weightage to each word of Training set and testing sets. Importantly, same transformation function is applied on the both training and testing data sets to constitute the similar features in both sets as validation of machine learning models can be rightly tested.

## V. EVALUATION OF MACHINE LEARNING MODELS

On transforming the training and testing data sets to required format with rightly assigned weightages, machine learning models have to be built to run on the training data and then to evaluate the efficiency of the models on identifying the spam comments. In this process, four machine learning models are chosen to conduct the proposed task. They are,

- Logistic Regression
- Random Forest Classifier
- Support Vector Machine
- Artificial Neural Network

### a. Logistic Regression:

Logistic Regression is used here as it is generally used to predict the happening of an event with respect to the depending factors. Logic Regression uses Logistic function as given in (4) that models quantities’ growth rate over the function of other quantity.

$$f(x) = \frac{L}{1 + e^{-k(x - x_0)}} \qquad (4)$$

Where, $x_0$ = x value of curve's midpoint
L = curve's maximum value
k = logistic growth rate

Logistic regression is binary model that provides result in the form - whether a word is a spam word or not i.e. 1 belongs to spam and 0 belongs to ham. Logic regression in python is accomplished using *SciKit Logistic Regression* model. This model offer various parameters to configure the model as per the requirement. Some of the default parameters are l2 penalty, ***inverse regularization strength of 1 (C)***, ***lbfgs solver algorithm*** etc. and by default regularization is applied on the model. Regularization aids in reducing the overfitting of the data by adding corresponding penalty and adds any benefits if the data is underfit. Lower (C) values corresponds to higher regularization strength i.e. minimizing the error as low as possible to provide the best fit model. On running the Logistic Regression on the trained data with ***Inverse regularization strength C = 1***, max iterations of ***100*** and solver algorithm of ***lbfgs*** and then testing it on the test data, **accuracy of 92.60%** is obtained.

b. Random Forest Classifier:

Random Forest Classifier (RMF) is an ensemble of decision trees and built on the basis of supervised algorithm as shown in Fig 3. Usually, overfitting is the issue in the machine learning models as the models fits the training data too close to the set of data points. This issue happens when the model understands the noise too closely. Usually, to prevent overfitting, regularization is added or the nodes in the hidden layers are diminished. But RMF does it by default as the group of decision trees run altogether.

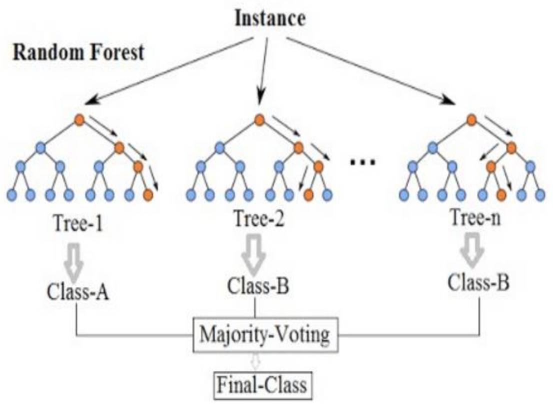


Fig. 3. Architecture of RMF

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i)) \qquad (5)$$

Where, C – number of classes totally
p (i) – Probability of picking a data in class i

In the project, ***RandomForestClassifier*** model of SciKit is used to build the model and run on the training samples. Parameters that are configured to build the RMF model are **'gini'** impurity function given in equation (5) that is used to measure the split with 100 trees in the forest, max-features is taken as square root function of all the n-features considered in the data set to decide the best split possible. The model is trained with 1563 comments as training samples with their corresponding weight and class values. The same is tested against testing sample set of 393 comments. The best accuracy achieved is **94.13%.**

c. Support Vector Machine (SVM):

Support Vector Machine (SVM) is a Linear Supervised model for classification and regression. SVM algorithm works on the principle of separating classes of data such that each class consists of similar elements and distinct elements are places in the different classes by analyzing the features and comparing it with different samples in the training set. Separating classes of data is achieved by margins, hyper planes and support vectors. In the context of SVM, margin correlates to area between the two classes of data. There might be some data that could be present between the areas of separation and are called outliers, those belong to either of the classes. To the estimate the proximity of outliers with respect either of the classes, hyperplane is drawn in the area of margin. Larger the margin area, higher the probability of better partition of classes and accordingly better likelihood of classification.

In the project, ***SciKit SVM*** machine model library is used to implement Support Vector Machine model and is configured by specifying regularization parameter as 100, ***radial basis function*** (rbf) as the kernel, kernel coefficient of gamma = 1 and **one-vs-rest** (ovr) is used as a decision function.

***Kernels*** aid the better classification of data by deciding best separation and hence the hyperplane. In this project, weightages assigned to each term extracted from the comments have to be classified into two classes. Here, the words picked from the comments are huge and randomly used over the comments – which might result difficult of differentiating the classes linearly. In such cases Kernel functions provides the way of finding the hyperplanes by mapping the data to n-dimensional space but calculating the coordinates in an actual rather than in higher dimensional space. Hence reduces the complexity of computation from ***O* (*n*$^2$) *to O* (*n*).**

***One-vs-rest (ovr) decision function*** is employed which tags labels for each data to determine whether it belongs to any class or not. Then each class is compared against other classes of data. Then, when a test data is picked to compare, it provides a decision whether it belongs to any classes and if then the class index. This model is trained on testing samples of 1563 comments and tested against 393 comments and results in accuracy of **93.33%.** As shown in fig 4, validation curve of SVM over the range of gamma values is plotted to observe that there is only small overfit as the gamma value increases.

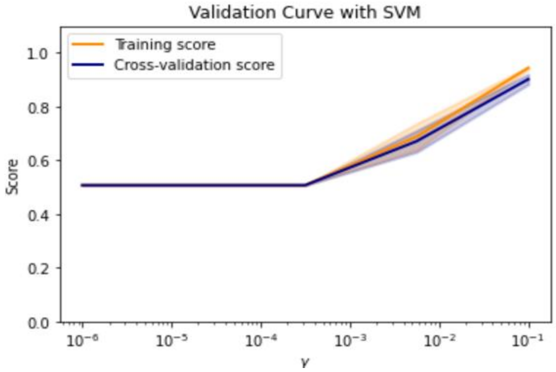


Fig. 4. Validation curve of SVM with gamma =1

d. Artificial Neural Network (ANN):

Artificial Neural Network is a supervised machine learning model that works on the composition and structure of the neural network embedded. Neural network structure resembles the human brain in both architecture and functioning. ANN consists of multiple layers like – Input, Output and hidden layers as shown in Fig. 5. The core functionality and efficiency of the neural network depends on the learning potential of the hidden layers configured. These layers consists of nodes that accomplishes that functionality of neurons in human brain. As there are numerous neurons in human brain, there will be numerous nodes in hidden layers. These nodes are designed to gain and refine their knowledge of the data in each iteration.

For the project, ***Multi-Layer Perceptron*** (MLP) model of SciKit is employed. MLP is a neural network model that linearly classifies the data by attaching a weight component on the inputs and bias component to construct a more flexible MLP model as in equation (6). On each iteration, weights are reassigned and neural network learns the data well through Backpropagation. It is method of regularly modifying the weight values to depreciate the error in classification.

$$y = \varphi(\sum_{i=1}^{n} w_i x_i + b) = \varphi(\mathbf{w}^T \mathbf{x} + b) \qquad (6)$$

Where, w – weight vectors, x – input vectors, b – bias and **φ** – non-linear function

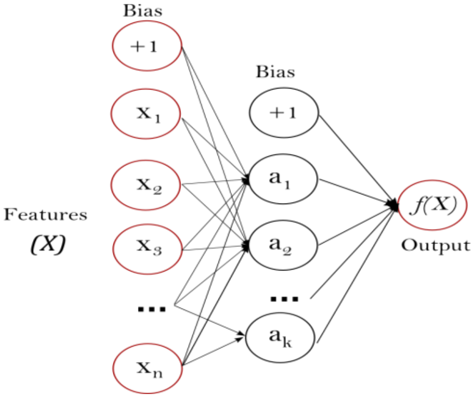


Fig. 5. Architecture of Neural Network

In this work, four hidden layers of each 20 nodes, maximum iteration count of 5000, and **rectified linear unit** (relu) function as activation parameter and **Adaptive Moment Estimation** (Adam) as the optimizer. It is an optimization of stochastic function – that measures the adaptive learning rates for distinctive features. This model was trained by inputting the training sample of 1563 comments and later tested against testing subset of 393 comments to provide the best accuracy of **92.85%.**

## VI. Summary and Conclusions

From the outset of data acquisition to lastly, studying the various machine learning models, understanding their training configuration, its effects and subsequently its performance in identifying the comments as spam or ham has been discussed in the paper. In the ‘related work’ section, other existing methods used in the other areas for spam detection and its inadequacy in adapting to YouTube spam detection is mentioned.

Logistic Regression, Random Forest Classifier, Support Vector Machine and Multi-Layer Perceptron models are used for the work. The accuracy of each of the machine learning model is calculated and Random Forest Classifier model provides the best accuracy of 94.13%. RMF is known to provide best results as aggregation of decision trees form the forest and unbiased estimate is furnished based on the model voting method, also ‘gini’ impurity function used in the RMF, **computes the gini gain and reduces the error that sequentially follows with best split and hence the better accuracy**.

On working with this project, I learnt the process of feature extraction using bag-of-words and tf-idf techniques. The heart of the matter topics of tokenizing, counting and vectorization of data is inspected and understood. Then, learnt the nitty-gritty of working with machine learning models and implementing each of them in python using SciKit libraries. Most importantly, I understood the principles of machine learning models and their configuration parameters. I learnt how the **selection of right regularization, impurity function, kernels, solver function, decision function and optimizers affect the training and eventually efficiency of the model**.

## VII. References

[1] https://hbr.org/2012/05/three-myths-about-customer-eng

[2] https://www.ncbi.nlm.nih.gov/pubmed/16248713

[3] https://www.factslides.com/s-YouTube

[4] https://theglobepost.com/2018/06/01/online-incivility-social-media/

[5] http://www.dt.fee.unicamp.br/~tiago//youtubespamcollection/

[6] https://medium.com/

[7] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011

[8] S. Orourke, Comment Spam Definition, November 2017, [online]Available: https://www.marketingterms.comldictionary/commentspaml.

[9] 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)*, Antalya, 2018, pp. 1-5.

[10] C. Rădulescu, M. Dinsoreanu and R. Potolea, "Identification of spam comments using natural language processing techniques," *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj Napoca, 2014, pp. 29-35.

[11] C. Huang, A. F. Molisch, R. He, R. Wang, P. Tang and Z. Zhong, "Machine-Learning-Based Data Processing Techniques for Vehicle-to-Vehicle Channel Modeling," in *IEEE Communications Magazine*, vol. 57, no. 11, pp. 109-115, November 2019.

[12] S. Mittal and O. P. Sangwan, "Big Data Analytics using Machine Learning Techniques," *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2019, pp. 203-207.

[13] N. Singh, D. P. Singh and B. Pant, "A Comprehensive Study of Big Data Machine Learning Approaches and Challenges," *2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS)*, Jammu, 2017, pp. 80-85.

[14] Jong Myoung Kim, Zae Myung Kim and Kwangjo Kim, "An approach to spam comment detection through domain-independent features," *2016 International Conference on Big Data and Smart Computing (BigComp)*, Hong Kong, 2016, pp. 273-276.

[15] N-Gram Assisted Youtube Spam Comment Detection, Shreyas Aiyar , Nisha P Shetty, International Conference on Computational Intelligence and Data Science (ICCIDS 2018)

[16] 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), Antalya, 2018, pp. 1-5.

[17] https://www.theguardian.com/technology/2013/nov/26/youtube-spam-comments-google-plus

[18] Kantchelian, J. Ma, L. Huang, S. Afroz, A. Joseph, J. D. Tygar, Robust detection of comment spam using entropy rate, in: Proceedings of the 5th ACM Workshop on Security and Artificial Intelligence, AISec ’12, ACM, New York, NY, USA, 2012, pp. 59–70. doi:10.1145/2381896.2381907.

[19] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, and M. Gonc¸alves, “Detecting spammers and content promoters in online video social networks,” in Proceedings - 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, 2009, pp. 620–627

[20] J. M. Campanha, J. V. Lochter, and T. A. Almeida, “Detecc ¸˜ ao autom´atica de spammers em redes sociais,” in Anais do XI Encontro Nacional de Inteligˆencia Artificial e Computacional (ENIAC’14),S ˜ ao Carlos, Brazil, 2014.

[21] M. McCord, M. Chuah, Spam Detection on Twitter Using Traditional Classifiers, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 175–186. doi:10.1007/978-3-642-23496-5 13.

[22] S. Choi, A. Segev, Finding informative comments for video viewing, in: 2016 IEEE International Conference on Big Data (Big Data), 2016, pp. 2457–2465. doi:10.1109/BigData.2016.7840882

[23] Tulio C. Alberto, Johannes V. Lochter, Tiago A. Almeida, TubeSpam: Comment Spam Filtering on YouTube, Federal University of Sao Carlos – UFSCar