Spam Detection in YouTube Comments Using Machine Learning

Tasin Mohammad
Department of Computer Science and Engineering
School of Data and Sciences
Brac University
Dhaka, Bangladesh
tasinms.bd@gmail.com

Annajiat Alim Rasel
Department of Computer Science and Engineering
School of Data and Sciences
Brac University
Dhaka, Bangladesh
annajiat@gmail.com

I. Introduction

YouTube has become one of the most popular websites for sharing and viewing video content online. With over a billion users and billions of comments posted each year, YouTube provides an open platform for people to engage in discussions about video content. However, the open nature of YouTube comments also makes them prone to spam. Spam comments can include irrelevant advertisements, malicious links, repetitive content and other unsolicited information. This can significantly degrade the quality of discussions and affect user experience on the platform. Detecting and filtering out spam comments has thus emerged as an important challenge for YouTube.

In this paper, we explore the application of machine learning techniques to detect spam in YouTube comments. Machine learning methods have shown promising results for text classification tasks across different domains. By learning to distinguish linguistic patterns in spam versus genuine comments, machine learning models can be employed for effectively automating YouTube spam detection. We present a systematic investigation of some core machine learning approaches, including Bernoulli Naive Bayes, Random Forests, and Support Vector Machines for this purpose. We curate a dataset of almost 2000 YouTube comments labeled as spam or genuine. We evaluate multiple classification algorithms over this dataset based on predictive performance metrics. Our experiments reveal insights into the most discerning features and most accurate machine learning models that can enhance YouTube spam detection significantly with a high degree of automation. The techniques proposed could be extended to identify and filter spam comments from other user-generated content platforms as well.

II. Literature Review

In literature, there is a substantial amount of research related to spam detection using different machine learning models. Trivedi et al. examines different machine learning classifiers for detecting spam emails [1]. Spam emails are an increasing nuisance, accounting for 70% of business emails by some estimates. They overwhelm inboxes, consume bandwidth and storage, and compromise security. There is a need for effective spam filtering methods to mitigate these issues. The paper examines Bayesian, Naive Bayes, SVM, decision tree, and boosted versions of Bayesian and Naive Bayes classifiers. It uses feature selection to reduce dimensionality and noise. The Enron email dataset is used due to its realistically complex spam examples. The key contribution is the comprehensive comparison of classifiers on metrics of accuracy, false positives, and training time. This provides guidance on the proper selection of machine learning techniques for real-world spam filtering based on performance requirements.

Kumar et al. provides a good introduction to the problem of email spam, defining it as ”using email to send unsolicited emails or advertising emails to a group of recipients without their permission” [2]. It notes that with the rapid growth of internet users, email spam is also increasing, and that spammers use it for illegal activities like phishing and fraud. The authors highlight that spam wastes storage, time, and reduces message speed. They state that while automated filtering is the most effective approach, spammers can still bypass these applications, making improved spam detection methods necessary. The authors use the Spam.csv dataset from Kaggle containing 5573 spam and non-spam emails for training machine learning models. They also create additional test datasets with 574 to 1001 emails to evaluate model performance on unseen data. The authors describe their overall experimental methodology covering data preprocessing tasks like cleaning, integration and reduction. For spam classification, they experiment with 8 supervised machine learning algorithms - Naive Bayes, SVM, Decision Trees, KNN, Random Forests, AdaBoost, Bagging and neural networks. Different configurations of feature extraction and hyperparameter tuning are tested for each method. The results compare all methods on metrics like accuracy, precision, recall, and F1-score. Key findings show Multinomial Naive Bayes has the best performance, but has limitations related to conditional independence assumptions. Ensemble methods like AdaBoost and Random Forests are also shown to be useful for leveraging multiple decision models. In conclusion, the authors summarize that their approach provides an effective spam filtering solution that categorizes emails based on content rather than sender metadata. They also outline several worthwhile extensions such as incorporating domain whitelisting and testing larger email corpora.

Sharmin et al. introduces the problem of spam threats affecting online social networks due to their open nature [3]. Spammers post unwanted content like ads, phishing links, fraud information etc. to promote their agendas. Detecting and filtering out such spam comments on social media platforms like YouTube is critical to provide good user experience. The main goal of the paper is to categorize YouTube comments as spam or non-spam using machine learning text classification techniques. Additionally, it aims to compare performance of ensemble and single classifiers. The methodology involves data collection, preprocessing, feature extraction, model building with various ML classifiers, followed by evaluation. Classifiers used were Naive Bayes, KNN, Bagging (ensemble method), and SVM. Evaluation metrics were Accuracy, precision, recall, F1-score, and MCC. Testing done via 10-fold cross validation. Accuracy over 80% achieved by Naive Bayes & Bagging on most datasets. Ensemble classifier Bagging outperforms individual classifiers on most benchmarks. MCC also highest for Bagging indicating good balance between false positives and negatives.

Kontsewaya et al. provides background on the problem of email spam, stating that over 85% of emails received by users today are spam [4]. Different types of spam are discussed, including advertising, phishing attempts, and Nigerian prince scams. The authors note that manual analysis of spam is impractical given the large volumes of data, and machine learning techniques can provide highly accurate spam classification. The aim of the paper is to evaluate different machine learning algorithms for detecting spam in order to create a more intelligent spam detection system. Six classification algorithms are selected: Naive Bayes, KNN, SVM, logistic regression, decision trees, and random forests. The authors use a natural language processing approach, analyzing the text content of emails to detect spam. They describe the typical machine learning workflow of data cleaning and preprocessing, model training, testing, and evaluation. Accuracy, precision, recall, F1, and ROC area are selected as evaluation metrics. Hyperparameter optimization is conducted for all models except Naive Bayes. The algorithms are trained and tested on a publicly available labeled dataset of 5728 emails from Kaggle. The data is split 80/20 into train and test sets. Various performance metrics are computed for each model, with Naive Bayes and logistic regression achieving the best results - up to 99% accuracy. In conclusion, the authors found Naive Bayes and logistic regression to perform the best for the spam detection task. They suggest these models could be combined or enhanced to create a more intelligent spam filtering system. Limitations and future work are not explicitly discussed. The study provides a useful comparative evaluation of different machine learning algorithms for the important real-world application of spam detection. It is reasonably well motivated and designed, but the brevity of the paper limits thoroughness. Expanding the literature review, datasets, experiments, and discussion of limitations would strengthen the work. Overall it makes a good contribution to the application area.

III. Methodology

A. Data Collection

The primary dataset used for this research is the comprised of 1956 rows of YouTube comments. The comments were labeled as spam (1) or not spam (0). The dataset was collected from Kaggle and the comments within the dataset are from music videos of popular artists which were uploaded to YouTube. The dataset is fairly balanced with 1005 spam comments and 951 comments which are not spam.

B. Data Visualization

*1) **Data Distribution**:* We initially visualized the distribution of spam and not spam comments in our dataset, where it was observes that the data was relatively balanced and thus did not require any additional sampling. Figure 1 visualizes the distribution of spam and not spam classes in the dataset.

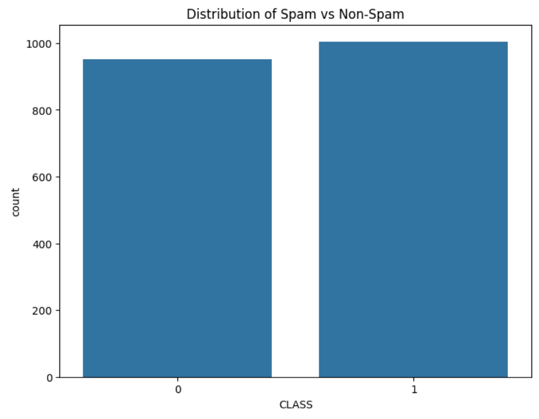

Fig. 1. Spam vs Not Spam

*2) **Word Cloud**:* Using a word cloud we visualised the most common words associated with spam comments in the dataset. Here we found out that *check*, *subscribe*, and *video* had the closest association with spam comments. Figure 2 show the word cloud for spam comments.

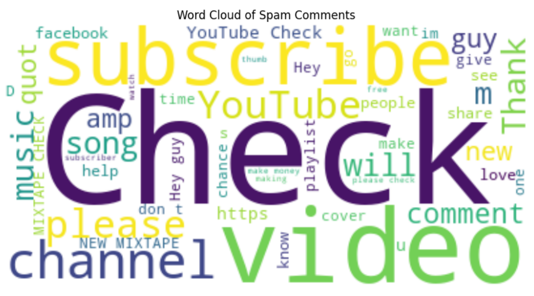

Fig. 2. Word Cloud For Spam Comments

C. Data Preprocessing

In order to prepare the data for spam classification we used count vectorization. This is a technique that converts text into numeric feature vectors based on the frequency of words. A count vectorizer works in the following way:

- It tokenizes the text by splitting it into separate words/tokens.
- It then builds a vocabulary of all the unique words. This vocabulary becomes the features.
- For each comment, it counts how many times each word in the vocabulary appears. This generates a numeric vector with each element representing the count for each word.

D. Models

3 models were trained and evaluated for our project. Each model underwent training with different hyperparameters on the validation set, and the results were compared with the objective of identifying the most precise and appropriate settings.

*1) **Bernoulli Naive Bayes**:* Bernoulli Naive Bayes is a classification algorithm that is well-suited for text classification tasks like spam detection.

The Naive Bayes family of classifiers work by calculating probabilities to make predictions. They use Bayes Theorem to calculate the probability that a given text belongs to a certain category (like spam or not spam).

The Bernoulli version specifically works with binary/boolean features. That means rather than indicating term frequency counts like some other Naive Bayes variants, the Bernoulli model just indicates absence/presence of keywords.

We chose this algorithm because it performs well with sparse binary/boolean data and can handle irrelevant words in comments well.

$$P(y|X) = \frac{P(X|y) \times P(y)}{P(X)} \tag{1}$$

*2) **Random Forest**:* A Random Forest consists of an ensemble of decision trees. Each individual decision tree is trained on a random subset of the available data features. This results in trees that are diverse and capture different aspects useful for classification.

During prediction, unseen comments get classified by passing them through each tree. Each tree votes for a particular class (e.g spam or not spam), and the forest chooses the classification with the most votes.

We chose this model because it can capture non-linear interactions between features. YouTube comments have complex dependencies and patterns that individual trees can model. It avoids overfitting compared to single trees. The ensemble encapsulates a rich and diverse set of classification rules. And it handles missing values gracefully. Spam detection still works fine even with incomplete data.

*3) **SVM**:* A Support Vector Machine aims to find the optimal hyperplane that maximizes the margin between different classes in the feature space. The key idea behind SVMs is to find the optimal boundary between different classes of data points. For YouTube comments, we would find the boundary separating likely spam comments from non-spam.

SVMs are powerful for spam classification because they are effective in high dimensional spaces. Comments have thousands of words/features that can be challenging for other algorithms. But SVMs handle these well. Moreover, it is robust to noise. Spam detection must sift through irrelevant words or typos. SVMs focus only on critical data points.

References

[1] S. Trivedi, "A study of machine learning classifiers for spam detection," pp. 176–180, 09 2016.
[2] N. Kumar, S. Sonowal, and Nishant, "Email spam detection using machine learning algorithms," pp. 108–113, 07 2020.
[3] S. Sharmin and Z. Zaman, "Spam detection in social media employing machine learning tool for text mining," pp. 137–142, 12 2017.
[4] Y. Kontsewaya, E. Antonov, and A. Artamonov, "Evaluating the effectiveness of machine learning methods for spam detection," vol. 190, pp. 479–486, 07 2021.