# Real Time YouTube video classification utilizing Structured Streaming and Machine Learning

Ananya Gopalakrishna
Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas
Richardson, Texas, USA
axg220262@utdallas.edu

Md Shahir Zaoad
Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas
Richardson, Texas, USA
mxz230002@utdallas.edu

Sandesh Gowda Shirgavara Ganesha
Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas
Richardson, Texas, USA
sxs220535@utdallas.edu

Suhas Ramesh
Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas
Richardson, Texas, USA
sxr220188@utdallas.edu

***Abstract*— YouTube, the leading video-sharing platform, is the source of information, news, entertainment, and knowledge for a wide variety of people. On average 3.7 million new videos are uploaded on YouTube per day which roughly covers the whole spectrum of human diversity and culture. Even though the YouTube algorithm personalizes video suggestions based on user preferences, there is a concern that this very algorithm can exploit user preferences to maximize platform engagement. This often leads to unnecessary user involvement with the application. To address this issue, this research aims at the classification of YouTube videos in real-time to deliver personalized content based on user preference. Leveraging Spark Structured Streaming this study handles the plethora of real-time YouTube content. An Artificial Neural Network (ANN) Multilayer Perceptron is incorporated for the classification task. The layered architecture of Neural Networks provides superiority over the traditional ML approaches. Moreover, the fewer number of parameters (weights and biases) of ANN make it a suitable candidate for Streaming applications than the deep learning approaches.**

***Keywords—YouTube video analysis, Streaming application, Machine Learning, Artificial Neural Network***

## I. Introduction

YouTube is among the leading platforms to provide global connectivity with around 2.5 billion monthly active users, a number expected only to rise. It serves as a critical tool for fostering relations among different demographics, providing world-class education, variety of information, and riches of entertainments. However, fueled by YouTube algorithm it can become a double-edged sword as this platform can become significantly addictive to its users, where he/she falls in the rabbit-hole of endlessly scrolling through videos/shorts [2 - 4]. Consequently, extensive use of social media platforms like YouTube, not only hampers one's productivity but also contributes to mental distress. The studies, [1], [5-6] show the correlation between extensive use of YouTube and mental health, especially in young demographics.

This research aims to solve the aforementioned issues utilizing a simple but intuitive approach – an alternative to the YouTube algorithm. The proposed solution is a streaming application that will utilize ANN to classify YouTube content in real time to suggest users' videos of their interest. This will work as a middleman between the user and YouTube, hence, instead of endlessly searching for content directly on YouTube, the user will only consume content suggested by our streaming application. The streaming application will monitor YouTube in real time to fetch videos being uploaded, titles of which will be stored as a Kafka topic. Followed by the pre-processing step this raw text will be sent to the trained ANN model for prediction of corresponding class. Finally, the videos from target classes will be presented to the user.

One of the crucial aspects of the system is the ANN classifier. It is the core engine for predicting classes of a YouTube video, hence it needs to be accurate while at the same time lightweight for the proper utilization with the real-time streaming application. Here lies the benefit of utilizing an ANN classifier. First of all, it is significantly more accurate than traditional ML approaches (Support Vector Machine, Logistic Regression, or Random Forest) on a task of this scale. Secondly, it is lightweight compared to deep learning approaches, for example, LSTM, and CNN as it uses relatively fewer parameters. As a result, it performs well with streaming applications with limited resources. Another crucial aspect is the training of the ANN classifier. We extensively searched popular online dataset resources, Kaggle, and Hugging Face library to find a suitable YouTube text classification dataset[1]. Preprocessing, and vectorization followed by word embedding have been performed to transform it into a suitable representation for the target model. Next, paramGrid was utilized for hyper parameter tuning followed by actual training and evaluation of the model. After rigorous experimentation, we finally created the classifier model that performed up to the intended standard.

## II. Problem description

Our model is utilized with streaming applications to predict the classes of YouTube videos in real time that acts as a middleman

[1] https://personal.utdallas.edu/~axg220262/Youtube1.csv

between YouTube and the user for efficient video retrieval. To sum up, the following are the key contributions of this project,

- Development of a real-time streaming application using spark structured streaming.
- Design, and development of a Multiplayer Perceptron in Map Reduce technology for YouTube video classification.
- Deployment of the model within the streaming application

## III. Background work

Big Data platforms like YouTube with its massive amount of information are a gold mine for research and analysis to extract hidden trends and crucial insight on certain demographics, world politics, peace, and many others. The past decade has seen much noteworthy research concerning YouTube data. The work could be broadly categorized into two categories – Analysis of Batched Data, and Analysis of Streaming Data. The former one is more saturated in terms of state-of-the-art research while the latter one is relatively unexplored.

### *A. Analysis of Batched YouTube Data*

The research in this area has gained a lot of attention in the last few years which ranges from content filtration, and isolation to establishment of Quality of Experience (QoE). The author of [7] presents an ensemble approach that comprises both a supervised and an unsupervised learning approach to classify YouTube content into one of six emotion categories - (happiness, anger, disgust, fear, sadness, and surprise). In another work [8], the research was aimed towards inappropriate content detection and classification. Here, the author proposes a novel deep-learning model to classify inappropriate content on YouTube data. The proposed model is a stack of a pre-trained CNN, EfficientNet-B7 followed by a Bidirectional LSTM with attention mechanism. Researchers have also explored the classification of YouTube videos from the title, description texts, and search terms. In [9] YouTube videos were classified into different categories based on search terms. The research compared the implementation of the system on three different machine learning techniques - SVM, Random Forest, and Naive Bayes. Naïve Bayes outperformed the other models with a 5-8% gain in accuracy. In another work [10], authors used a traditional Machine Learning approach, Random Forest to classify YouTube videos into several categories based on Title and Description text. There has also been work similar to [16] where authors performed statistical analysis on YouTube videos. They found that videos have strong correlations among themselves that could be leveraged to enhance service quality. Analysis has not only been performed on the content itself but also on the Meta data to enhance the user's QoE [11-15].

### *B. Analysis of streaming YouTube Data*

However, most of the research discussed in the previous section utilized historical YouTube data. While these can provide important knowledge and insight into the information being curated within a certain interval, they lack handling real-time data. One of the key concerns with real-time Big Data is to handle and analyze them as they become available rather than waiting for a later time when they are no longer useful. Hence, it is crucial to gain insight into the data in real-time as they become available. Leveraging Big Data streaming technologies, namely, Apache Structured Streaming [17], Apache Kafka, Amazon Kinesis, Redis Streams, and many others real-time analysis can be performed seamlessly and fruitfully.

Do, P., Pham, P. and Phan, T. [18] utilized spark streaming to detect Harmful comments from various social media including YouTube. The proposed system utilizes various big data technologies under the Apache family to extract, process, and analyze text and image data to successfully detect harmful content across social media platforms. In another work [19], the authors utilized publicly available social media data to identify real-world disaster events. Latent Dirichlet Allocation, a topic modeling approach based on Apache Spark is used to propose the disaster event detection framework. The proposed model outperformed the existing statistical methods with an accuracy of 96%. Researchers also ventured into implementing Machine Learning algorithms in a distributed system for faster training and utilizing models in real time. Herodotou, H., Chatzakou, D. and Kourtellis, N. [20] proposed a classification model built on top of Spark that detects online aggression in real-time. They incorporated real-time ML algorithms (Hoeffding Trees, Adaptive Random Forests, and Streaming Logistic Regression), to iteratively train the data as they become available. For this purpose, they used Twitter Streaming API to incrementally fetch the data and corresponding labels. However, it is not quite possible to find such an API/system that provides labeled data in a streaming fashion.

Even though contemporary work addressed real-time analysis of YouTube content, there has hardly been work that ventured into the classification of YouTube video in real-time to associate it with user preference. A handful of research [21-22] explored the generation of top YouTube categories based on real-time video analysis. For this purpose, they relied on the YouTube APIs and their default categories. The primary concern with this approach is that YouTube has a handful of categories that are highly generalized and are automatically assigned to a video which is not accurate in many cases. Users can specific video categories however, the option is relatively hidden within an overwhelming interface, and hence, creators often overlook this part while uploading videos.

To address this shortcoming, our research aims to propose a novel end-to-end streaming application leveraging Artificial Neural Networks to classify YouTube videos in real-time. Instead of relying on the default classes of YouTube API, we trained our ANN model on a labeled dataset of massive scale to predict the most accurate category for a particular class. Contemporary ML-based streaming applications mostly rely on traditional ML approaches. While it is efficient to build, they only perform well with small datasets in a static environment.

They are no longer viable while dealing with massive amounts of real-time data. As a result, we chose the ANN classifier which also performs significantly better than deep learning alternatives when there is a resource limitations on the streaming technologies.

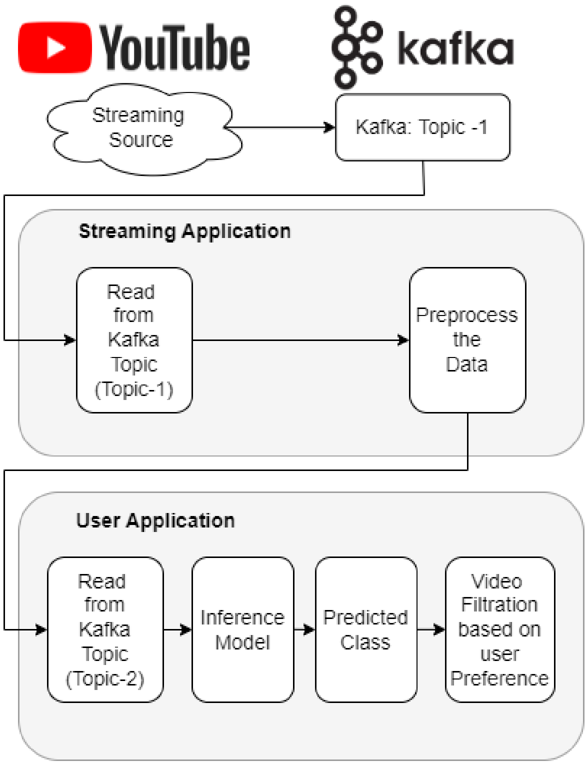

Fig. 1. System Architecture

We designed and developed the ANN model in a parallelized setup to utilize the processing power of distributed computing. Finally, the proposed system can successfully extract videos based on user preferences.

## IV. Methodology

The working principle of the system is divided into two subsections - Streaming Application, and User Application which is depicted on the Fig.1. Initially, the streaming YouTube data is sent to a Kafka topic for temporary storage. Our Streaming Application then starts by reading form the temporary Kafka topic. It preprocess the data and sends it to the User Application. The User Application uses the processed input to generate corresponding class prediction for the purpose of intended filtration of YouTube videos based on user preferences. The entire system is implemented on Docker for efficient resource management and organization of modules. Following is the details concerning each modules,

### A. *Streaming Application*

This section starts by establishing connection to specific Kafka broker following by reading data from Kafka topic. The input data is real-time video titles from YouTube. The input then goes through the following of preprocessing steps,

*1) Hyperlink Removal:* Regular expression is used to identify strings that starts with ‘http’ and are replaced with empty string.

*2) Word Frequency Threshold:* Words that occurred at least 3 times are considered as a significant word for model training. Words below this threshold are discarded.

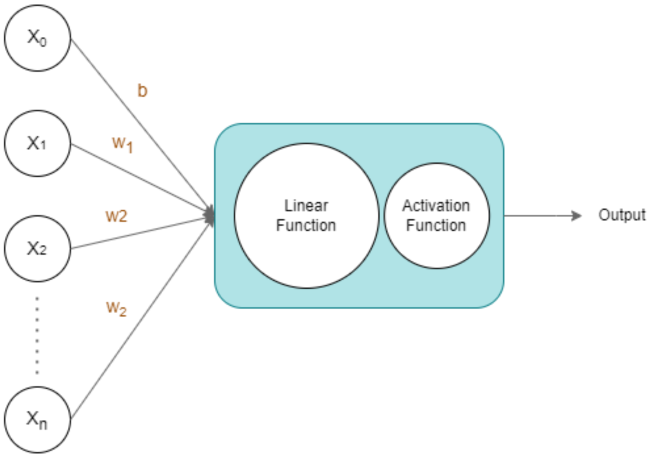

Fig. 2. Artificial Neuron

*3) Tokenization:* In this step each sentence is split into a list of tokens using Tokenizer class of Spark’s MLlib library.

*4) Stop Words Removal:* Stop words are common filler words that contributes little to the actual sematics of a sentence. Hence, in this step we get rid of stop words.

*5) Vectorizer.* This step takes the output of the previous step and generates vectorized represnetaion.

All these preprocessing step has been implemented leveraging Aapche Saprk’s distributed architecture. Finally the processed output and corresponding lable is written into another Kafka topic.

### B. *User Application*

This reads data from streaming application. It then uses the preprocessed input and passes it to the trained ANN model to generate the prediction class. However, before we can use this inference model, we have to train it. The next section delves deeper into this.

### C. *Artificial Neural Network*

Multilayer Perceptron (MLP) is the intended classifier model for this system. The basic building block of MLP is a neuron, inspired my human brain. Each neuron takes weighed inputs and biases and generates outputs leveraging the activation function. Fig.2 depicts such a neuron where $X_i$’s are inputs, $w_i$’s are weights and b is the bias. These are combined with a linear function which is the passed through the activation function.

In MLP these neurons are stacked in layered architecture. Connection is formed between neurons from different layers whereas there is no connection among neurons in the same layer.

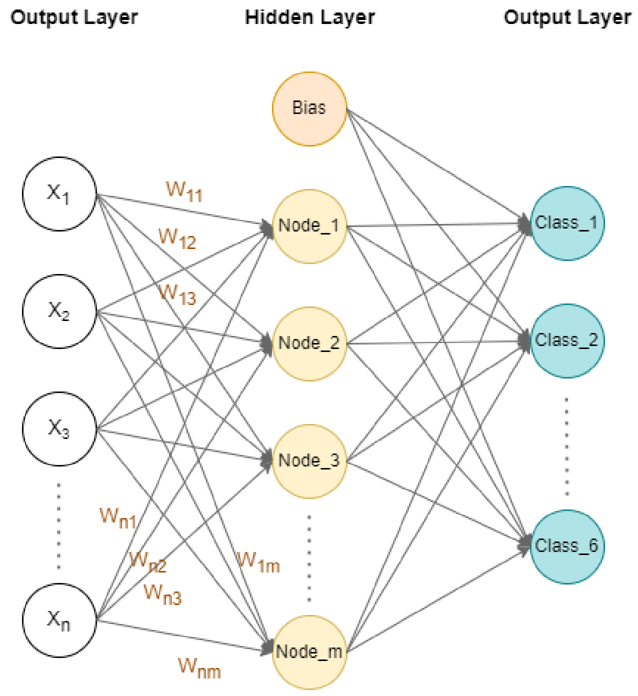

Fig. 3. Multi-Layer Perceptron

These connection helps to learn features from the input which ultimately helps to predict output on an unseen data. Fig.3. represents a MLP classifier with an input layer, an output layer, and a hidden layer. In the input layer there are n neurons each represents a word representation form a sentence. Now, similar to a single neuron, calculation is performed in each neuron in a layer-by-layer basis and inputs are transformed into outputs.

One of the crucial aspect of neural network is activation function that is responsible for generating the output of each neuron. In our implementation we used following activation function,

*1) Softmax:* It generates probability distribution of our output classes. The probability of individual classes sum to 1. This is a popular choice for multiclass classification problem. The class with the highest porbability value is the corresponding predicted class for an input.

*2) ReLU:* In hidden layers Rectified Linear Unit (ReLU) activation function is used. It sticks to lienar transforamtion for positive values while it outputs zero for negative values. It introduces non-linearity to the model that helps the model to learn complex, non-linear realtionship from the data, hence is a popular choice for hidden layers.

### *D. Dataset*

In order to learn the model needs data, a large amount of them. To accomplish this we extensively searched popular dataset repositories, namely Kaggle, and Hugging Face library and finally decided on a dataset from Kaggle with 10.0 out of 10.0 usability. The data set has 11,212 rows with four columns (Title, videourl, Category, and Description). There are six distinct classes (Food, manufacturing, History, travel blog, Science&Technology, and Art&Music). This dataset also underwent same types of preprocessing used in the inference phase. Finally, the model is ready for training.

### *E. Training*

We trained the model in a distributed system, Apache Spark. It is a parallel processing system, and hence allowed an efficient and speedy training. For training we used 70% data, the rest has been saved for evaluation. In this process the data has been split while preserving their initial class distribution. Moreover, we trained the model on different combination of key hyper parameters to identify the most suitable one for our use case. Two configurations were tested: (10,5) and (20,10), indicating the number of neurons in each hidden layer. The (10,5) configuration has fewer neurons in each layer compared to (20,10), implying a simpler model architecture. Two learning rates were tested: 0.03 and 0.3. A higher learning rate (0.3) implies larger updates to the model's weights during training compared to the smaller learning rate (0.03). Three different numbers of epochs were used: 20, 50, and 100. An epoch represents one complete pass through the entire training dataset during the training phase.

## V. Result and Discussion

Table.I represents the summary of the hyper-parameter tuning with corresponding accuracy,

TABLE I. HYPER-PARAMETER TUNING

| Combinations | Hidden Layers | Learning Rate | Epochs | Accuracy |
|---|---|---|---|---|
| C1 | 10,5 | 0.03 | 100 | 97.66 |
| C2 | 20,10 | 0.03 | 100 | 97.74 |
| C3 | 10,5 | 0.03 | 50 | 96.31 |
| C4 | 20,10 | 0.03 | 50 | 96.46 |
| C5 | 10,5 | 0.03 | 20 | 82.52 |
| C6 | 20,10 | 0.03 | 20 | 82.97 |
| C7 | 10,5 | 0.3 | 20 | 82.57 |
| C8 | 10,5 | 0.3 | 100 | 97.66 |
| C9 | 10,5 | 0.3 | 50 | 96.55 |
| C10 | 20,10 | 0.3 | 20 | 82.97 |
| C11 | 20,10 | 0.3 | 100 | 97.74 |
| C12 | 20,10 | 0.3 | 50 | 96.46 |

The model with the highest accuracy has been selected as the target model which is the 2nd row in the Table I with an accuracy of 97.74% for 20+10=30 hidden layer nodes, learning rate of

0.03 and epochs of 100. From the table, we also see that epoch has the most effect on the model's performance rather than learning rate and nodes in the hidden layers.

The model is evaluated on the 30% of the data from the dataset. First, we processed the data with the same preprocessing steps mentioned above. This processed data are then sent to the inference model that generates an output, the prediction class. These are then evaluated with the actual class labels to measure the performance of the MLP classifier. We used the following popular evaluation metrics for multiclass classification problem,

*1) Accuracy:* It paints an overall picture of the model's perfroamnce, the total percentage of correct classficiation among all the classfication. However, it fails to extract nuances, for example if the classes in the dataset are imbalanced accuracy won't be able to detect it.

*2) Recall:* It is the true positive rate. It measures the proportional of positive cases identified out of all the positive ones.

$$\text{Recall} = \text{True Positive}/\text{True Positive}+\text{False Negative} \tag{1}$$

*3) Precision:* It measures the proportion of true positive cases out of all the positive cases predicted by the model.

$$\text{Precision} = \text{True Positive}/\text{True Positive}+\text{False Positive} \tag{2}$$

*4) F1 score:* It is the harmonic mean of precistion and recall where it will assign weiths based on their values. Lower value would have more weights assign to it, hence lowerin the F1 score. It is an unbaised representation of the model's true performance.

$$\text{F1 score} = 2 \text{ x Precision x Recall} / (\text{Precision} + \text{Recall}) \tag{3}$$

The MulticlassClassificationEvaluator, and MulticlassMetrics classes of Spark MLlib have been utilized to perform evaluation of our model. Following figures are the summary of the result,

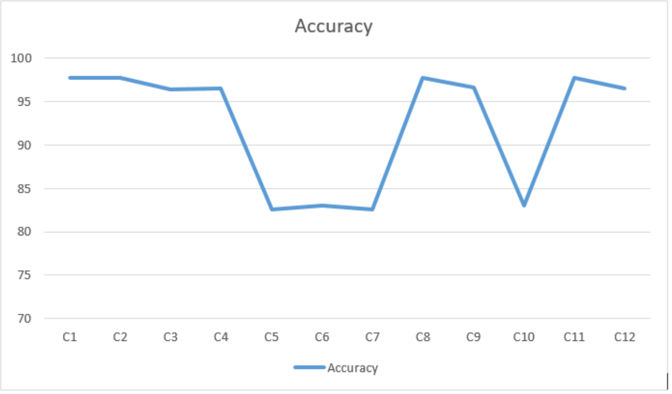

Fig. 4. Accuracy comparison for 12 configurations

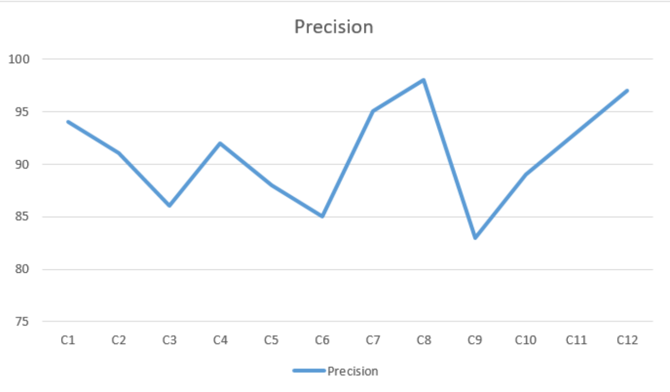

Fig. 5. Precision comparison for 12 configurations

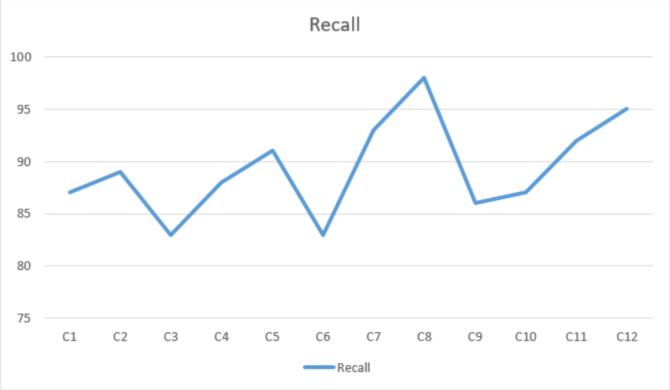

Fig. 6. Recall comparison for 12 configurations

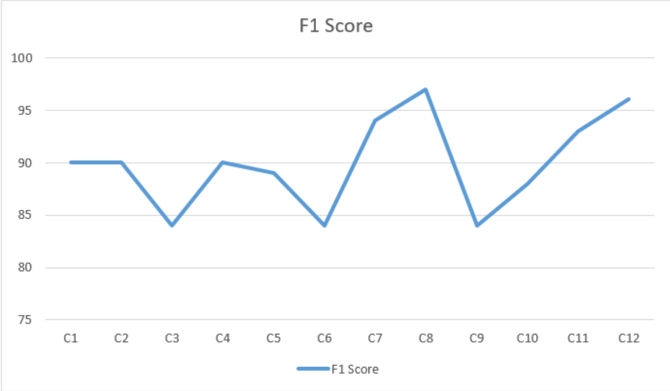

Fig. 7. F1 score comparison for 12 configurations

From the Fig.4. to Fig.7 we see that Precision and recall scores vary across configurations but generally show similar trends as accuracy. F1 scores, being a balance between precision and recall, also exhibit similar trends, with higher values indicating better overall performance. It is clear that the accuracy of the model is actual indicator of the model's performance as the other metrics conforms to its values. Hence, choosing the best model based on accuracy was the valid choice.

## VI. Conclusion

This research took the endeavor of presenting a solution to the social media, especially YouTube addiction. Arguably this platform is implemented in such a manner that exploits user psychology to increase the App usage. To tackle this situation we proposed a middle man between YouTube and user that will selectively extract videos from the YouTube based on user preference and will only present those selective videos to the user. To achieve this we utilized Spark streaming application coupled with Machine Learning. The proposed model achieves an accuracy of 97.74% in classifying videos in one of the six distinct categories. Finally, the system successfully overcomes the aforementioned concern. In future we plan to build a user interface around the core functionality of the middle man software to provide user a more authentic experience.

## REFERENCES

[1] Balcombe, L. and De Leo, D., 2023, April. The Impact of YouTube on Loneliness and Mental Health. In Informatics (Vol. 10, No. 2, p. 39). MDPI.

[2] Klobas, J.E., McGill, T.J., Moghavvemi, S. and Paramanathan, T., 2019. Problematic and extensive YouTube use: First hand reports. Online Information Review, 43(2), pp.265-282.

[3] Moghavvemi, S., Sulaiman, A.B., Jaafar, N.I.B. and Kasem, N., 2017, July. Facebook and YouTube addiction: The usage pattern of Malaysian students. In 2017 international conference on research and innovation in information systems (ICRIIS) (pp. 1-6). IEEE.

[4] Balakrishnan, J. and Griffiths, M.D., 2017. Social media addiction: What is the role of content in YouTube?. Journal of behavioral addictions, 6(3), pp.364-377.

[5] De Bérail, P., Guillon, M. and Bungener, C., 2019. The relations between YouTube addiction, social anxiety and parasocial relationships with YouTubers: A moderated-mediation model based on a cognitive-behavioral framework. Computers in Human Behavior, 99, pp.190-204.

[6] Sevim, S., Gumus, D. and Kizil, M., 2024. The relationship between social media addiction and emotional appetite: a cross sectional study among young adults in Turkey. Public Health Nutrition, pp.1-13.

[7] Chen, Y.L., Chang, C.L. and Yeh, C.S., 2017. Emotion classification of YouTube videos. Decision Support Systems, 101, pp.40-50.

[8] Yousaf, K. and Nawaz, T., 2022. A deep learning-based approach for inappropriate content detection and classification of youtube videos. IEEE Access, 10, pp.16283-16298.

[9] Amanda, R. and Negara, E.S., 2020. Analysis and implementation machine learning for youtube data classification by comparing the performance of classification algorithms. Jurnal Online Informatika, 5(1), pp.61-72.

[10] Kalra, G.S., Kathuria, R.S. and Kumar, A., 2019, October. Youtube video classification based on title and description text. In 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 74-79). IEEE.

[11] Wamser, F., Seufert, M., Casas, P., Irmer, R., Tran-Gia, P. and Schatz, R., 2015, June. YoMoApp: A tool for analyzing QoE of YouTube HTTP adaptive streaming in mobile networks. In 2015 European Conference on Networks and Communications (EuCNC) (pp. 239-243). IEEE.

[12] Ragimova, K., Loginov, V. and Khorov, E., 2019, June. Analysis of YouTube dash traffic. In 2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom) (pp. 1-5). IEEE.

[13] Seufert, M., Casas, P., Wamser, F., Wehner, N., Schatz, R. and Tran-Gia, P., 2016, July. Application-layer monitoring of QoE parameters for mobile YouTube video streaming in the field. In 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE) (pp. 411-416). IEEE.

[14] Krishnappa, D.K., Bhat, D. and Zink, M., 2013, October. DASHing YouTube: An analysis of using DASH in YouTube video service. In 38th Annual IEEE Conference on Local Computer Networks (pp. 407-415). IEEE.

[15] Seufert, M., Wehner, N., Wamser, F., Casas, P., D'Alconzo, A. and Tran-Gia, P., 2017, May. Unsupervised QoE field study for mobile YouTube video streaming with YoMoApp. In 2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX) (pp. 1-6). IEEE.

[16] Cheng, X., Dale, C. and Liu, J., 2008, June. Statistics and social network of youtube videos. In 2008 16th Interntional Workshop on Quality of Service (pp. 229-238). IEEE.

[17] Armbrust, M., Das, T., Torres, J., Yavuz, B., Zhu, S., Xin, R., Ghodsi, A., Stoica, I. and Zaharia, M., 2018, May. Structured streaming: A declarative api for real-time applications in apache spark. In Proceedings of the 2018 International Conference on Management of Data (pp. 601-613).

[18] Do, P., Pham, P. and Phan, T., 2020. Some Research Issues of Harmful and Violent Content Filtering for Social Networks in the Context of Large-Scale and Streaming Data with Apache Spark. Recent Advances in Security, Privacy, and Trust for Internet of Things (IoT) and Cyber-Physical Systems (CPS), pp.249-272.

[19] Bhuvaneswari, A., Jayanthi, R. and Meena, A.L., 2021, August. Improving crisis event detection rate in online social networks twitter stream using apache spark. In Journal of physics: Conference series (Vol. 1950, No. 1, p. 012077). IOP Publishing.

[20] Herodotou, H., Chatzakou, D. and Kourtellis, N., 2020, December. A streaming machine learning framework for online aggression detection on Twitter. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 5056-5067). IEEE.

[21] Sirisha, Y., Parimala, K.S. and Jyothi, A.K., 2017. You Tube Data Analysis Using Hadoop Technologies Hive. Smart and Sustainable Initiatives in Natural Sciences and Engineering, 1, pp.10-16.

[22] Reddy, J.M., Attuluri, A., Kolli, A., Sakib, N., Shahriar, H. and Cuzzocrea, A., 2022, December. A Crowd Source System for YouTube Big Data Analytics: Unpacking Values from Data Sprawl. In 2022 IEEE International Conference on Big Data (Big Data) (pp. 5451-5457). IEEE.