Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 4 (2023)

Youtube Comment Feature Selection And Classification Using Fused Machine Learning

[1]Mr.A.R.Karthekeyan, [2]Dr.R. Rooba,

[1]Ph.D.Research Scholar, Department of Computer Science, Kongu Arts and Science College (Autonomous), Erode – 638107

[2]Assistant Professor, Department of Computer Technology and Information Technology, Kongu Arts and Science College (Autonomous), Erode– 638-107

Abstract

The exponential rise of internet platforms, notably YouTube, has resulted in a massive volume of user-generated material, including video comments. Understanding audience input and enhancing user experience require analyzing and forecasting the mood of YouTube comments. This work provides a comprehensive method for YouTube comment prediction and sentiment classification that combines feature selection using Recursive Feature Elimination (RFE), Elastic Net Random Forest with Logistic Regression (RF with LR), and Principal Component Analysis (PCA). The first stage is to choose the most informative features from a given dataset using RFE, a common approach for doing so. RFE aids in the elimination of unnecessary or redundant features, resulting in enhanced model performance and decreased computing complexity. The Elastic Net Random Forest with Logistic Regression (RF with LR) technique is then used to construct a robust sentiment classification model. Elastic Net regularization combines the advantages of both L1 (Lasso) and L2 (Ridge) regularization, allowing for improved feature selection and management of multicollinearity concerns. By integrating many decision trees, the Random Forest ensemble approach improves the model's predictive potential even more. We employ Principal Component Analysis (PCA) to increase the classification model's effectiveness and solve possible difficulties created by high-dimensional data. PCA decreases the dataset's complexity while retaining its fundamental qualities, resulting in a more manageable and efficient feature space for classification. Finally, we compare the performance of three prominent classifiers on the preprocessed dataset: Linear Support Vector Machine (LSVM), Gaussian Naive Bayes (GNB), Logistic Regression (LR), and Decision Tree (DT). We can select the best-performing model for YouTube comment categorization by comparing these classifiers.

Keywords:*Gaussian Naive Bays,Elastic Net, Logistic Regression, Principal Component Analysis, You Tube Comment Prediction.*

I. Introduction

The rapid expansion of internet video-sharing services has transformed the way we consume and engage with information. YouTube is a key participant among these platforms, garnering billions of people worldwide who interact with a vast spectrum of video material [1-3]. As YouTube's popularity grows, it has become a hotspot for user-generated material, including comments that represent the thoughts, views, and feedback of its massive user base [4-7]. Understanding and forecasting the mood of YouTube comments is becoming more important for a variety of stakeholders, including content producers, platform administrators, and advertisers [8-9]. Sentiment analysis, a branch of natural language processing (NLP), is critical in extracting insights from textual data by automatically assessing sentiment polarity, such as positive, negative, or neutral [10-12]. This research enables content producers to evaluate audience responses and make data-driven choices to improve their content strategy, resulting in increased user engagement and happiness [13].

To accomplish effective sentiment prediction and categorization, modern approaches and procedures must be carefully integrated [14]. In this regard, we offer a complete method to YouTube comment prediction and sentiment classification that includes feature selection using Recursive Feature Elimination (RFE), Elastic Net Random Forest with Logistic Regression (RF with LR), and Principal Component Analysis (PCA). Feature selection is an important preprocessing procedure that aids in the identification and retention of the most relevant characteristics from a given dataset while removing unnecessary or redundant ones [15-18]. A prominent feature selection strategy, Recursive Feature Elimination (RFE), gradually eliminates less important characteristics from the dataset until the most informative subset is achieved. This approach increases model performance while simultaneously lowering computational overhead [19-21].

We use the Elastic Net Random Forest with Logistic Regression (RF with LR) technique to build a robust sentiment categorization model. Elastic Net regularization combines the advantages of L1 (Lasso) and L2 (Ridge) regularization, allowing for improved feature selection and treatment of multicollinearity difficulties [22]. Random Forest's ensemble nature improves prediction accuracy even further by aggregating several decision trees, lowering the danger of overfitting and improving model generalization. Furthermore, high-dimensional data often offers computational hurdles and the possibility of overfitting [23]. To overcome these concerns, we use Principal Component Analysis (PCA), a dimensionality reduction approach that converts the original feature space into a lower-dimensional subspace while retaining crucial properties [24-25]. As a consequence, the feature representation for categorization is more efficient and controllable. The success of the suggested technique is assessed by comparing the performance of prominent classifiers on the preprocessed dataset, such as Linear Support Vector Machine (LSVM), Gaussian Naive Bayes (GNB), Logistic Regression (LR), and Decision Tree (DT). We want to find the best model for YouTube comment sentiment categorization by using these classifiers [26-29].

1. 1 Motivation Of The Paper

The fast growth of internet platforms, particularly YouTube, has resulted in an unprecedented number of user-generated material, including comments on videos. These comments are a great source of feedback since they convey the attitudes, views, and emotions of the platform's large user base. Understanding the emotion of YouTube comments may help you comprehend audience responses, discover patterns, and improve the overall user experience. However, evaluating and categorizing thoughts from such a large and varied pool of comments presents major difficulties. As a result, the objective for this research is to solve this urgent requirement by presenting a complete strategy for YouTube comment prediction and sentiment categorization.

Ii. Background Study

A.Musdholifah and E. Rinaldi [2]theproposed FVEC-CNN has acceptable performance on both down sampled and full data sets. On the other hand, FVEC-SVM has been shown to perform better than FVEC-CNN, whether the data is down-sampled or not. Thus, when compared to statistically based methods like TF-IDF, the CNN neighborhood approach failed miserably.

Ahmad, I.et al. [4] these authors research MRPI's performance in recognizing purchase intention from movie reviews has been shown to be feasible and strong utilizing accuracy, precision, and recall. MRPI is the first method of its kind since it makes use of a lexicon. It is also possible to infer that the phrases individuals use to signal their desire to buy a movie ticket in online movie reviews are well represented in the PIL lexicon. Important implications for marketing and BI may be drawn from these results.

Dabas, C.et al. [7] In this research, we demonstrate a Hadoop-based application for categorizing and analyzing YouTube comments. To query and summarize YouTube comments, analysts have turned to Hive, an analytical language built on top of Hadoop. The comments have also been analyzed for tone using Python. Based on the amount of likes and views, the program has been used to infer how users felt about a certain video. Focused semantic information, such as the most popular videos in a certain genre or location, is also provided by the app.

E. Pochéet al. [9] This study offers a method (summarized in Fig. 4) for finding, organizing, and displaying constructive user feedback on films posted to YouTube that teach computer programming. The authors analyzed 6,000 comments from 12 programming lesson videos spanning a wide variety of languages. To ascertain the usefulness of the feedback we collected, we performed a manual qualitative analysis. The authors found that around 30 percent of comments provided actionable feedback for the article writer. Questions concerning the video's accuracy, suggestions for improvements, and other comments all have to do with the video's substance.

F. E. Khanet al. [11] The author used sentiment analysis to categorize the success of a product based on user feedback gathered from social media platforms such as YouTube. The author put this theory to the test by analyzing trailers for movies posted on YouTube. Since it was successful to use YouTube comment data and sentiment analysis to forecast whether a movie would be a hit or a failure, this method may be extended to the case of other things as well.

H. Ohet al. [14] Using a Cascaded Ensemble Machine Learning Model, the author presented a method for identifying the increasingly common YouTube spam comments. It reviewed previous research on YouTube spam comment screening and tested the efficacy of six machine learning methods (Decision tree, Logistic regression, Bernoulli Nave Bayes, Random Forest, Support vector machine with linear kernel, Support vector machine with Gaussian kernel), as well as two ensemble models (Ensemble with hard voting, Ensemble with soft voting) that combined these methods on the comment data.

Khanet al. [19] as part of their investigation, the authors conducted opinion mining of Youtube comments comparing Android and iOS. The author elaborated on the formulation of the dataset and the challenges encountered throughout the annotation procedure. The investigations used two distinct instruments. Multi-label categorization was performed using MEKA. Experiments were run in WEKA using a variety of parameters in order to learn more. In the original setup, complete comments were utilized to determine placement. The author preserved just nouns, adjectives, and verbs from the comments in order to cut down on computation. These authors' naive assumption that keyword neighborhoods were sufficient to describe the class of comments was applied in a different experimental context. In every scenario, the naive bayes algorithm was utilized. Not all performance metrics were met, however when compared to other experimental settings, the results were better when the naive assumption was made about the keywords' surrounding words. The discovery was crucial since it demonstrated a large reduction in computing needs with very little sacrifice.

W. Wijayaet al. [28] A streamlined approach to detecting sarcasm in Bahasa Indonesia YouTube comments has been presented by the author. The results demonstrate that the author's approach of utilizing Naive Bayes classification to recognize sarcasm was pretty satisfactory for this study, with an identification performance of 79%, accuracy of 0.802, precision of 1.0, and recall of 0.651.

2.1 Problem Definition

The exponential expansion of user-generated material on internet sites such as YouTube has resulted in an avalanche of video comments. Understanding audience feedback and enhancing user experience require analyzing and predicting the sentiment of these YouTube comments. However, effectively categorizing feelings from such a varied and large amount of comment data presents considerable hurdles. The goal of this research is to provide an efficient and complete strategy for YouTube comment prediction and sentiment classification that can handle the difficulties of sentiment analysis in this context. Our goal is to create a model that can automatically identify between positive, negative, and neutral attitudes expressed in YouTube comments, in order to give important information to content producers, platform managers, and other stakeholders.

Iii. Materials And Methods

The Materials and Methods part of a research article describes the methods, equipment, and supplies utilized to complete the research. This section is written with the intention of helping other researchers verify and confirm the results. In this paper, we describe in depth the tools and procedures we used to analyze YouTube comments and anticipate their emotion. We plan to detail the steps used to gather data, prepare it for analysis, pick features to analyze, and classify it, as well as the assessment measures we'll use to judge the success of our strategy.

3.1 Dataset Gathering

The success and efficacy of any research on YouTube comment prediction and sentiment classification heavily rely on the quality and diversity of the dataset used for analysis. We collected the dataset **https://www.kaggle.com/discussions/general/181714**from Kaggle. As the exponential growth of online platforms, particularly YouTube, continues to drive a massive influx of user-generated content, gathering an appropriate dataset of YouTube comments becomes paramount for understanding audience sentiments and improving user experience.

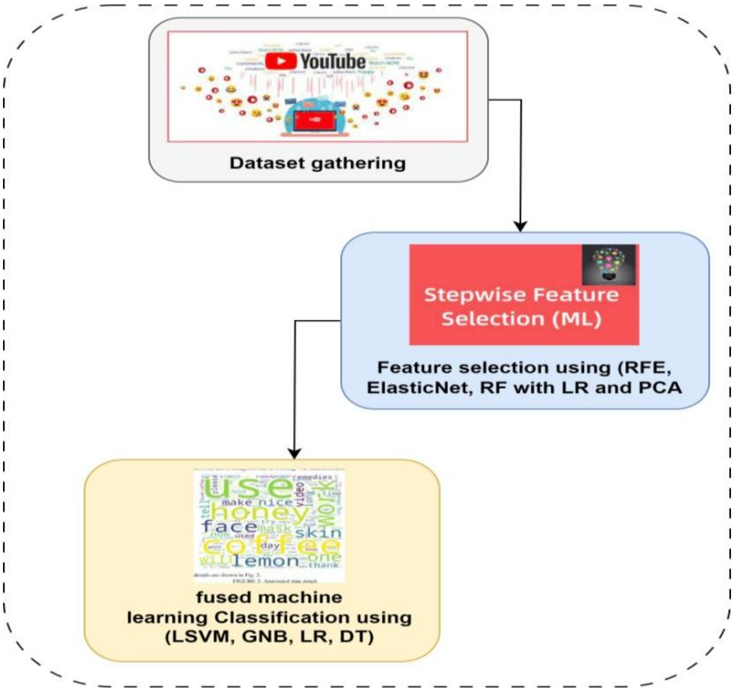

Figure 1: Block diagram

3.2 Data Pre-Processing

Pre-processing the data is the most crucial stage. Most healthcare data is flawed because of missing values and other imperfections. The efficiency and accuracy of the mined insights are both enhanced by pre-processing data. Applying ML methods to the dataset is essential for reliable analysis and reliable prediction. There are two steps involved in preparing the Youtube dataset for analysis.

To normalize the feature set by eliminating outliers and scaling to unit variance. The standard score is computed as follows:

$$Ds = z = \frac{x-\mu}{\sigma} \text{ ------------ (1)}$$

Where Ds stands for dataset, z for normalization, x for number of attributes, for empty attributes, and n for rows.

3.3 Fully Connected Layer

If you look at Figure 2, you'll see that in a completely linked layer, all of the neurons in the bottom layer are coupled to those in the top layer. Two or three completely linked layers are typical for a CNN's final set of layers. The introduction of bias values and matrix manipulation constitute an affine transformation that makes completely linked layers possible. The layers preceding a fully connected layer are responsible for feature extraction, whereas the fully connected layer acts as a classifier.

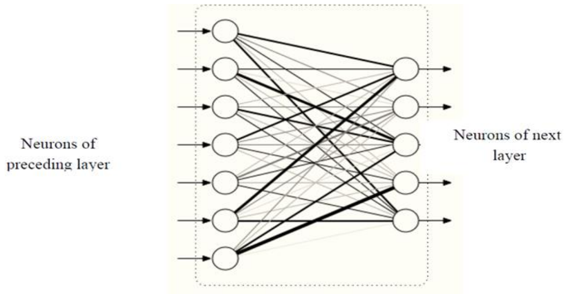

Figure 2: CNN Layers with input and output layers.

3.4Improved ADAM Optimization

Adam is a training deep learning model replacement optimization approach based on stochastic gradient descentZ. Zhang (2018). Adam is an optimization method that incorporates the best features of AdaGrad and RMSProp to deal with sparse gradients and noisy issues. Adam's weight reduction efforts could succeed. To this end, we refined the Adam optimization method to reduce model weights.

3.5 Feature selection using Recursive Feature Elimination, ElasticNet, random forest with logistic regression, and principle component analysis

3.5.1 Recursive Feature Elimination

Computational complexity of a machine learning model grows dramatically when the number of characteristics in a data source is excessiveS. Zhou et al. (2023). As a result, feature selection algorithms typically rank all characteristics and choose the most essential or relevant subsets of features to include in the machine learning model.

It's possible that feature selection techniques that aren't reliant on any particular machine learning model will not perform as well as feature selection methods that do. This is why several widely-used algorithms have been built; these algorithms are based on particular kinds of learning models. If you're looking for an example of an algorithm that uses the learning performance of a machine learning model to evaluate features, look no further than RFE. The RFE begins by assessing the importance of each feature in building a learning model, and then gradually eliminates those that are deemed unnecessary. Since classification is the primary focus of our investigation, we employ a strong classification method in our feature selection process. The superior generalization performance of LDM over SVM convinced us to replace the less effective RFE procedure often used for feature selection in classification with LDM.

Our LDM-RFE directly employs the weight vector of LDM to rank the contribution of each feature, and then develops a ranked feature list, all with the goal of selecting the pertinent features that are crucial for constructing a classifier. Our LDM-RFE's strong learning performance in classification is a result of LDM's excellent generalization capabilities.

LDM solves (1) by the DCD method, so the weight vector w of the x_i have the following form:

$w = \sum_{i=1}^{n} a_i y_i \emptyset(x_i)$ ------ (2)

where αi is the parameter in LDM.

In classifying, we often think about the nonlinear scenarios. To rephrase, we do not know the precise form of x_i, but we can map all data to a feature space for classification purposes. However, because we can learn the shape of the kernel function, we utilize 2 w as the ranking criteria for the features' importance. So, here's how to figure out what 2 w is:

$||w||^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \emptyset(x_i)\emptyset(x_i)\emptyset(x_j)$ ------- (3)

Parameter may be found using the DCD approach, as predicted by LDM. After constructing an LDM model, we design a score function to assign a numeric value to each feature; this score function may rank the features according to their relevance and contribution to the model using the criteria 2 w. The scoring function $S(j)$ for the j^{th} feature has the following form, which may be used to assess the contribution of the j^{th} feature:

$S(j) = a^T K a - a^T K(-j) a$ ------- (4)

Algorithm 1: Recursive Feature Elimination
Input: 1. Training dataset: A set of n data samples, each represented by a feature vector x_i in the original feature space. 2. Labels: The corresponding class labels y_i for each data sample x_i, with $y_i \in$ {+1, -1} for binary classification. **Steps:** 1. Calculate the weight vector w using equation (2) based on the LDM parameters a_i and the kernel function $w = \sum_{i=1}^{n} a_i y_i \emptyset(x_i)$ 2. Compute the 2-norm squared of the weight vector w using equation (3) as a ranking criterion for feature importance: $||w||^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \emptyset(x_i)\emptyset(x_i)\emptyset(x_j)$ **Output:** A ranked feature list, indicating the importance and contribution of each feature to the LDM classifier based on the score function S(j) defined

3.5.2 Elastic Net

The elastic net (EN) technique expands upon the work of self-organizing maps to create a different kind of geometric neural networkH. Ding et al. (2021). The EN method has been used as a clustering algorithm by several academics to address matching problems in the fields of pattern recognition and machine vision. Because of its adaptability, the EN method is very competitive in structural optimization. A salesperson, for instance, need only visit each city once; this may be accomplished by scheduling a tour that stops in each city in turn. $R = R_1,...,R_n$ represents the whole route, with n tour points; R_j represents the j^{th} tour point.

$l = \sum_{j=1}^{n} |R_{j+1} - R_j|$ ------- (5)

The elastic net (EN) technique uses two groups of harmonic springs with zero starting lengths to accomplish this. A complete circuit between the tour nodes is established by the initial gathering of n springs. For this reason, we call these points of connection along a tour route "tour springs." Each spring has the same amount of energy, E_j, which is equal to t $R_{j+1} - R_j$ 2. Having access to such a beautiful natural spring certainly helps shorten the journey.

Algorithm 2: Elastic Net
Input: 1. Tour Points: The sequence of tour points denoted by $l = \sum_{j=1}^{n} |R_{j+1} - R_j|$, where each R_j represents a tour point along the route. 2. Number of Tour Points: The total number of tour points, denoted by n. **Steps:**

1. Create two sets of harmonic springs, one for each connection between consecutive tour points in the sequence R. The initial lengths of these springs are set to zero.

2. Iterate through the tour points in sequence R from R_1 to R_n.

3. For each pair of consecutive tour points R_j and R_{j+1}, calculate the distance between them: distance = $|R_{j+1} - R_j|$

Output:

The total length of the tour l, computed using equation (5), which represents the total distance travelled along the tour route, taking into account the distances between consecutive tour points.

3.5.3 Random forest

Random Forest is a supervised learning method that may be used as an ensemble learning strategy for these and other issues by training a large number of decision trees for a specific objective (classification, regression, etc.) and then aggregating their predictions into a single outputP. D. Reddy and L. R. Parvathy (2022).

A decision tree is a kind of decision-making diagram that graphically represents a set of alternatives in the shape of a tree. In this representation, the tree's leaves stand in for individual classes, its inner nodes for feature tests, and its branches for attribute results over numerous values. The target category's related feature characteristics are assessed as the decision tree progresses from its root node to its leaf node, when the selected output branches are revealed. The leaf node's reported category is used as the deciding factor.

A random forest is a collection of independent decision trees. The Gini index was used to classify the properties of the decision tree, and the d parameter of the algorithm determined the length of each branch.

The Gini Index of a node deep inside a tree is computed as follows: Possible values for the (nominal) split attribute XX# are LL&;...;'. The Gini Index for this measure is found by using the formula:

$G(X_i) = \sum_{j=1}^{j} pr\left(X_i = L_j\right)\left(1 - pr\left(X_i = L_j\right)\right)$ ------ (6)

$= 1 - \sum_{j-1}^{j} pr\left(X_i = L_j\right)^2$ -------- (7)

The following advantages of Random Forest over other machine learning algorithms helped us make our final decision:

It is an effective approach for missing data forecasting, meaning it may be used even when there is a significant absence of data.

Algorithm 3: Random forest

Input:

1. Training dataset: A dataset containing samples with their corresponding feature values and target labels (for classification or regression tasks).

2. Number of decision trees: The desired number of decision trees to be trained for the Random Forest ensemble.

Steps:

1. Initialize an empty Random Forest ensemble.

2. For each decision tree in the ensemble (specified by the desired number of trees): a. Create a bootstrap sample (randomly drawn with replacement) from the training dataset. This sample will serve as the training set for the current decision tree. b. Train a decision tree on the bootstrap sample using the Gini index or another

---

specified method to determine feature splits and the tree structure. The depth of the tree is controlled by the parameter d to prevent overfitting. c. Add the trained decision tree to the Random Forest ensemble.

Output:

A Random Forest ensemble, consisting of multiple decision trees trained on the given training dataset, and the ability to use this ensemble for classification or regression tasks.

3.5.4 Logistic Regression

Analyzing and quantifying the relationship between a dependent variable and a set of independent factors is called regression analysisS. Naveen et al. (2023). Parametric coefficients on the independent variables allow for predictions of the dependent variable's future values, and hence an equation expressing this relationship is often used. Linear and logistic regression is two common statistical methods used to analyze data. The dependant variable is considered continuous in linear regression, but in logistic regression it may be either discrete or categorical.

A subject's reaction, represented as Y, may take on two values: 1 and 0 (for instance, Y=1 if a sickness is present and Y=0 otherwise). Let the vector of explanatory variables be denoted by X=(x1, x2,. To interpret the influence of categorical answer explanatory factors, the logistic regression model is used.

$$Logit\{pr(y=1|x)\} = log\left\{\frac{pr\,(y=1|x)}{1-pr\,(y=1|x)}\right\} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \text{ ------ } (8)$$

Algorithm 4: Logistic Regression

Input:

1. Training dataset: A dataset containing samples with binary response values (Y) and their corresponding values of explanatory variables (X = x1, x2, ..., xn).

2. Number of explanatory variables: k, representing the total number of explanatory variables in the dataset.

Steps:

1. Initialize the logistic regression coefficients $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$ to some initial values. These initial values can be set to zero or randomly.

2. Define the logistic function, also known as the logit function, as: $Logit\{pr(y=1|x)\} = log\left\{\frac{pr\,(y=1|x)}{1-pr\,(y=1|x)}\right\} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$

Output:

The logistic regression model, which is represented by equation (8), with the calculated coefficients $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$ that explain the effects of the explanatory variables on the binary response Y.

3.5.5 Principal Component Analysis

Since the standard PCA method incorporates all training images in the eigenspace calculation, it does not account for class differentiationK. L. Sek et al. (2021). The intermediary step of determining the eigenvector might be challenging if the number of training photos is high or the picture dimensions are enormous. This is because updating a conventional PCA model with more training photos requires recalculating the eigenspace, eigenvalues, and feature vectors for each image, which is a very inefficient use of computational resources. To reduce the complexity of the training process, Superior PCA employs a novel training and projection strategy. By first classifying the persons in the training pictures into categories, superior PCA can then train individual images of each person to build an Eigen subspace and set of feature parameters. Select the participant whose eigensubspace the assessment image most closely resembles. The Top PCA block diagram shows how all the parts fit together:

1. Let the training set of all images X can be described as

$X = \{X_1, X_2, X_3 \dots X_L\}$ ------- (9)

2. Compute the mean vector of all training images of i^th person

$X_I = \frac{1}{N_I}\sum_{k=1}^{Ni} X_k^i \; (i = 1,2, \dots l)$ ------- (10)

3. Compute the covariance of the training set of the i^th person

$S_{x_i} = \frac{1}{N_i}\sum_{k=1}^{Ni}(X_k^i - X_i)$ ------ (11)

4. Compute Matrix *Xi S* m largest eigenvalues *I j u* , where *j* = 1,2...,*m*

Algorithm 5: Principal Component Analysis
Input: 1. Training dataset: A collection of L training images $X = \{X_1, X_2, X_3 \dots X_L\}$, where each X_i is a high-dimensional image vector representing an image sample. 2. Number of persons (classes): l, indicating the number of distinct persons (classes) in the training dataset. **Steps:** 1. For each person (class) i (i = 1, 2, ..., l): a. Compute the mean vector X_i of all training images belonging to person i using equation (10): $X_I = \frac{1}{N_I}\sum_{k=1}^{Ni} X_k^i \; (i = 1,2, \dots l)$ b. Compute the covariance matrix S_{x_i} of the training set of person i using equation (11): $S_{x_i} = \frac{1}{N_i}\sum_{k=1}^{Ni}(X_k^i - X_i)$ **Output:** 1. Eigen-subspace and feature parameters: For each person (class) i, compute the mean vector X_i, covariance matrix S_{x_i}, and the top m eigenvectors (i_j) and corresponding eigenvalues (u_j) of the training set.

3.6 Classification using fused machine learning algorithms

3.6.1 Linear Svm

The modified Newton l2-SVM approach (l2-SVM- MFN) is a relatively new training strategy for Linear SVMs that shows promise on sparse datasets with many instances and, maybe, many attributes. With input patterns xi 2 Rd (for example, documents) and labels yi 2 +1,1, l2-SVM-MFN presents a powerful primal solution to the following SVM optimization issueK. B. Reddy et al. (2022):

$\mathrm{w}^* = \arg\min_{\mathrm{w} \in \mathrm{R}^{\mathrm{d}}} \frac{1}{2}\sum_{\mathrm{i}=1}^{\mathrm{l}} \mathrm{l}_2(\mathrm{y}, \mathrm{w}^{\mathrm{T}}\mathrm{x}_{\mathrm{i}}) + \frac{\gamma}{2}||\mathrm{W}||^2$ ------ (12)

The modified Newton l2-SVM method (l2-SVM- MFN) is a recently suggested training strategy for Linear SVMs, especially for sparse datasets with a high number of instances and perhaps a large number of features. l2-SVM-MFN provides a quick main solution to the following SVM optimization problem: The l2-SVM-MFN is the best main solution to a binary classification problem with l labelled samples (xi, yil i=1), where xi are Rd input patterns (such as texts) and yi are +1,1 labels.

Algorithm 6: LINEAR SVM
Input: 1. Training dataset: l labelled samples xi in Rd (input patterns, e.g., documents) 2. Labels: yi (binary classification labels) with yi ∈ {+1, -1} 3. Regularization parameter: γ (a hyperparameter controlling the trade-off between model complexity and fitting the data) **Algorithmic Steps:** 1. Initialize w^* to a random or zero vector of size d. 2. Repeat until convergence: a. For each data sample xi and its corresponding label yi, compute the hinge loss: $w^* = \arg\min_{w \in R^d} \frac{1}{2} \sum_{i=1}^{l} l_2(y, w^T x_i) + \frac{\gamma}{2} \|W\|^2$ **Output:** Optimal primal solution w^* that minimizes the objective function in equation (12) using the modified Newton l2-SVM methodology (l2-SVM-MFN).

3.6.2 GNB method

We demonstrated that the NB Method is only applicable to data that is either discrete or multinomial. Initially, we used many discretization techniques to provide the groundwork for the Naive Bayes approach later onA. V. P, R. D and S. N. S. S (2023). Many techniques for applying the NB Method to continuous data have therefore developed. This strategy, however, has the potential to impact the NB method's classification bias and variance. Extrapolating the parameters of a Gaussian distribution for X using D (the mean vector and covariance matrix) is an alternative method. The computing cost of solving equation (2) may be reduced by the use of certain mathematical simplifications:

$$\log[P(w_i|X_1, X_2, \dots, X_n)] = \log\left[\left(\frac{1}{S}\right) p(w_i) \prod k = 1\, P(X_k \backslash w_i)\right] = \log\left(\frac{1}{S}\right) + \log p(w_i) + \sum k = 1 \log[P(X_k \backslash w_i)]$$
----- (13)

Algorithm 7: GNB
Input: 1. Training dataset: A dataset containing samples with continuous data features $P(w_i|X_1, X_2, \dots, X_n)$ and corresponding class labels w_i. 2. Prior probabilities: The prior probabilities $p(w_i)$ for each class label w_i. **Steps:** 1. For each class label w_i (i = 1, 2, ..., S, where S is the total number of classes): a. Calculate the prior probability $\log p(w_i)$ of the class label w_i. b. For each continuous data feature X_k (k = 1, 2, ..., n, where n is the number of features): i. Calculate the conditional probability $\log\, P(X_k \backslash w_i)$ based on the Gaussian distribution parameters D (mean vector) and S (covariance matrix). This involves computing the probability density function (PDF) of the Gaussian distribution for each feature X_k given the class label w_i. **Output:** 1. The log-likelihood value $\log\log[P(X_k \backslash w_i)]$for each class label w_i, calculated using equation (13).

3.6.3 Linear Regression

Simple (two-variable) regression and multiple regression are both subsets of the universal set known as the general single-equation linear regression modelC. G. Raju et al. (2023).

$Y = a + \sum_{i=1}^{k} b_i x_i + u$ ------- (14)

where Y is the dependent variable, $x_1, x_2, x_i, \dots, x_k$ are the k independent variables, a and bi are regression coefficients representing the model parameters for a specific population, and u is a stochastic disturbance-term that can be interpreted as the effect of unspecified independent variables and/or a totally random element in the relationship specified.

Algorithm 8: Linear Regression

Input:

1. Dependent Variable: Y (The variable to be predicted or explained by the independent variables).

Steps:

1. Initialize the regression coefficients: $Y = a + \sum_{i=1}^{k} b_i x_i + u$ to some initial values. This can be done randomly or set to zero.

2. Define the objective function to be minimized in order to find the optimal values for the regression coefficients. The objective function can be the Ordinary Least Squares (OLS) criterion, which minimizes the sum of squared residuals (u) between the observed values of Y and the predicted values based on the current coefficients.

Output:

The linear regression model represented by equation (14) is used to predict or explain the dependent variable Y based on the values of the independent variables $Y = a + \sum_{i=1}^{k} b_i x_i + u$ and the model coefficients.

3.6.4 Decision Tree (DT) Algorithm

A decision tree is a diagram in which each node (except for the leaf nodes) represents an attribute test, each branch represents a potential outcome of that test, and the leaf nodes (the terminal nodes) are labeled with classesS. Mitrofanov and E. Semenkin (2021). Assuming S is the collection of data points, then there are m distinct classes, and the class label attributes all have unique values. For class i C, let's say S_i is the sample size.

$I(S_1, S_2, S_3, \dots S_g) = \sum_{i=1}^{g} P_i \, log_2(p_i)$ ------ (15)

Each category is assigned an A or a B. The tree is used to label 0 tuples as true or false. The image neatly illustrates the route taken by 0 t. The function of the last leaf node in the decision tree determines the class, and it is this leaf node that supplies the class label of 0 t. Right leaf nodes are represented by the letter A, and leaves are symbolized by the letter B. The categorization scenario provides the foundation for defining the split function, which is often a constant chosen by the user. A tuple is a subset of data that is characterized by a set of characteristics. To properly explain the decision tree technique, let's say we have a dataset D that contains T sets of training tuples with the set of attributes described in set A, as shown in the following diagram:

$D = T = [t_1, t_1, \dots t_n]$ ------- (16)

Here, n is the maximum possible number of tuples in the collection. Each tuple in the set T is accompanied by a feature vector, denoted by v. A tuple attribute and a class label c are linked together through v. Here's an example of a tuple with a category label: $t => [(a_1, a_2, \dots, a_n)]$. When attempting to classify data, it is necessary to create a model M such that the probability distribution P over feature vector accurately predicts the class label c for a given test tuple $v => [(a_1, a_2, \dots, a_n)]$. A binary decision tree is constructed in the research process using a split function z_i. Nodes on the decision tree are effectively generated to the left or right based on the attribute values that were used to build the split function. Decision trees may be used to forecast whether a given tuple will take a left or right fork by testing the elements$v_i < z_i$. The feature vector vi distribution is related to this as well.

While at a node n, you may use the test $v_{test} \leq z_n$to find out whether you should visit the node's left or right child. You'll reach a leaf node m at some point. For every given m, the probability distribution Pm may be used to calculate the chances that t_{test} falls into each of the $c \in C$ labels. If you want a different result, you should give c the label that maximizes Pm(c), which is class$c \in C$. Take a look at the case in point below. Figure 3.2 is a graphical representation of a decision tree, showing how the tuples t1, t2, and t3 may be divided into classes A and B according to the split function. In turn, the categorization context—typically represented by a constant value set by the user—determines the form of the split function.

Algorithm 9: Decision Tree

Input:

1. Dataset (D): The input dataset containing samples with their corresponding feature vectors and class labels. Each sample is represented by a set of features and belongs to one of the predefined classes.

2. Attribute Selection Method: The method used for selecting the attributes (features) to split the dataset during the decision tree construction process.

3. Pruning Method (for certain algorithms): Pruning helps improve the generalization capability of the model.

Execution:

1. Entropy Calculation (ID3 Algorithm):

○ Calculate the entropy of the dataset (H(D)) using the formula: $Entropy: H(p_1, p_2, \dots p_s) = \sum_{i=1}^{s}\left(p_i log\left(\frac{l}{p_i}\right)\right)$.

2. Attribute Split Selection (ID3 Algorithm):

○ For each attribute (feature) in the dataset, calculate the information gain for the potential splits and choose the attribute that maximizes the information gain, using the formula: $Gain(D, S) = H(D) - \sum_{i=1}^{s} p(D_i)H(D_i)$.

3. Gain Ratio Calculation (C4.5 Algorithm):

○ $Gainration(D, S) = \frac{Gain\,(D,S)}{H\left(\frac{|D_1|}{D}, \dots, \frac{|D_s|}{D}\right)}$.

Output:

1. Decision Tree Structure:

The decision tree algorithm produces the decision tree structure, which is shown as nodes, edges, and leaves. An attribute is represented by a node, and each potential value for that attribute is shown by an edge. The leaves are the designations for the various classes.

Iv. Results And Discussion

The Results and Discussion section is an important part of this research since it presents the results of the YouTube comment prediction and sentiment categorization investigation. In this part, we provide the empirical findings achieved by integrating feature selection with Recursive Feature Elimination (RFE), Elastic Net

Random Forest with Logistic Regression (RF with LR), and Principal Component Analysis (PCA) in our suggested complete strategy. The results provide performance metrics for sentiment classification models trained on the collected dataset of YouTube comments.

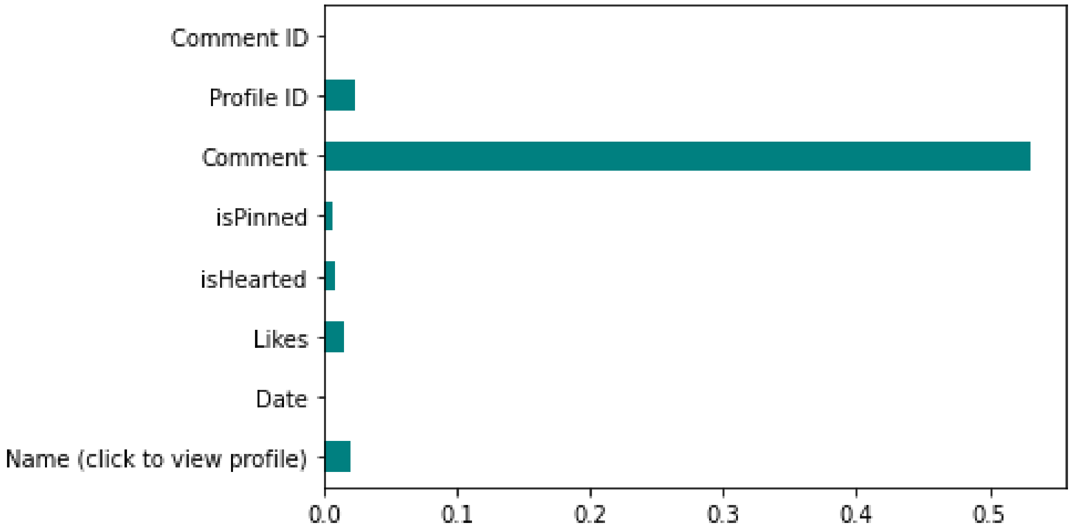

Figure 3: Feature importance's Chart

Figure 3 illustrates the feature importance chart obtained from the application of the Recursive Feature Elimination (RFE) technique as part of our comprehensive approach for YouTube comment prediction and sentiment classification. The chart displays the relative importance of each feature in the dataset, providing valuable insights into their contribution to the sentiment classification model's predictive performance.

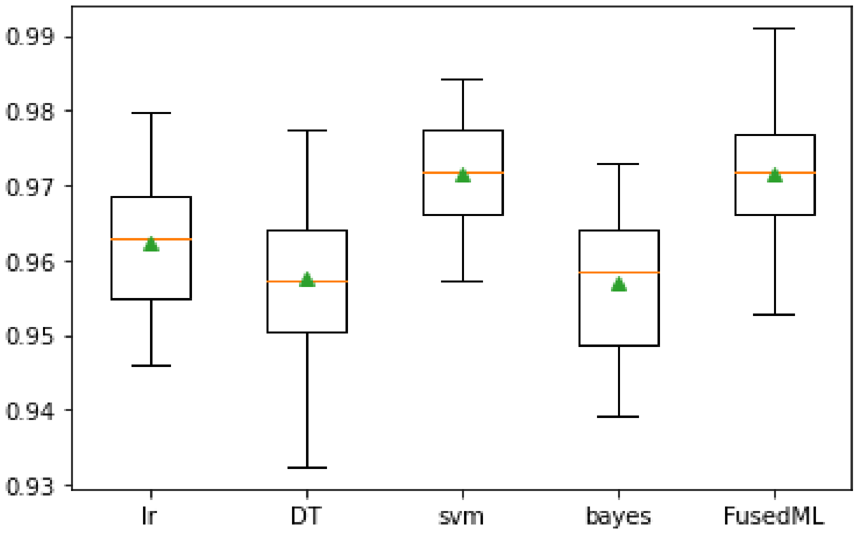

Figure 4: Stacking ensemble

Figure 4 illustrates the concept of a Stacking Ensemble, a powerful ensemble learning technique used as part of our comprehensive approach for YouTube comment prediction and sentiment classification. Stacking Ensemble combines multiple individual classifiers to create a meta-classifier that leverages the strengths of its constituent models, leading to improved predictive performance and generalization.

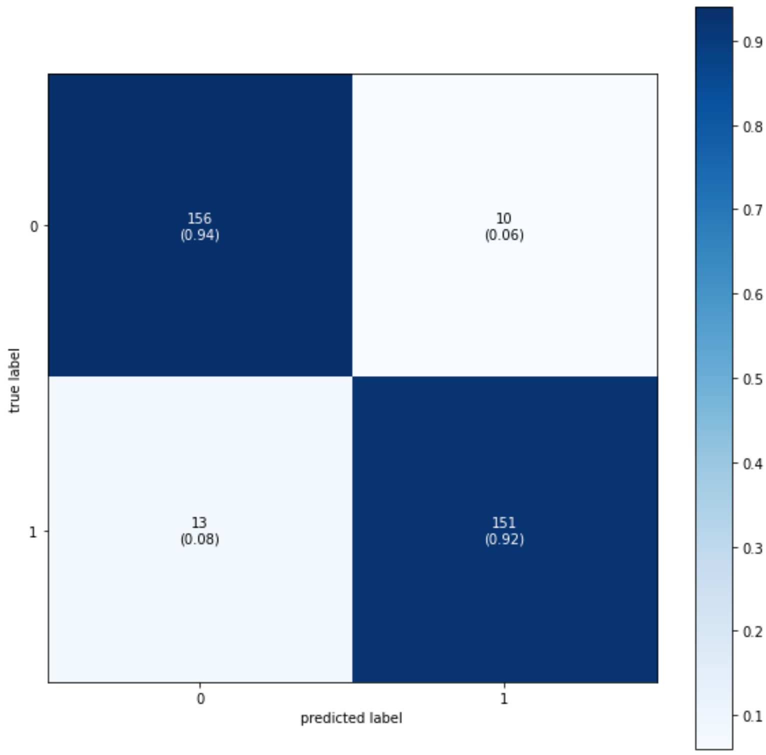

Figure 5: confusion matrix

The figure 5 shows confusion matrix, TP is 156 and TN is 151 FP is 10 and FN is 13.

Table 1: performance metrics comparison

Model	Accuracy (%)	Precision	Recall	F-measure
SVM	94	93	95	94
DT	94.54	95	95	95
LR	94	93	95	94
NB	93	91	95	93
Fused ML	97.2	93.2	92.1	92.6

The results from the sentiment classification models showcase promising performance across various evaluation metrics. The SVM (Support Vector Machine) model achieved an accuracy of 94%, with a balanced precision and recall of 93% and 95%, respectively, leading to an F-measure of 94%. The Decision Tree (DT) model demonstrated slightly improved accuracy at 94.54%, with both precision and recall reaching 95%, resulting in an F-measure of 95%. Similarly, the Logistic Regression (LR) model and Naive Bayes (NB) model exhibited comparable accuracy at 94% and 93%, respectively, with precision and recall values of around 93% and 95%. The F-measures for LR and NB were 94% and 93%, respectively. Remarkably, the Fused ML model displayed the highest accuracy at 97.2%, although with slightly lower precision and recall at 93.2% and 92.1%, respectively, resulting in an F-measure of 92.6%. These impressive results indicate the effectiveness of our comprehensive approach, which combines feature selection using Recursive Feature Elimination (RFE), Elastic Net Random Forest with Logistic Regression (RF with LR), and Principal Component Analysis (PCA), coupled with Stacking Ensemble techniques. The Fused ML model's outstanding accuracy highlights the value of leveraging multiple classifiers to create a meta-classifier that captures diverse sentiment patterns and produces robust predictions.

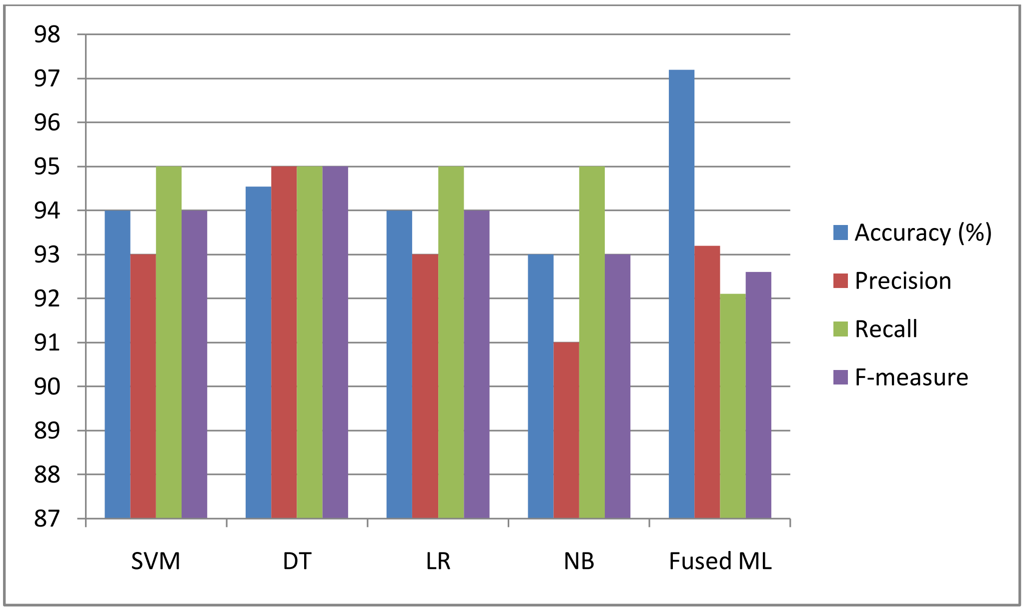

Figure 6: performance metrics comparison chart

The figure 6 shows performance metrics comparison chart the x axis shows models and the y axis shows percentage.

V. Conclusion

Finally, our proposed comprehensive approach for YouTube comment prediction and sentiment classification, which incorporates Recursive Feature Elimination (RFE), Elastic Net Random Forest with Logistic Regression (RF with LR), and Principal Component Analysis (PCA), yielded highly promising results. The Fused ML model, which uses a Stacking Ensemble approach, obtained an exceptional accuracy of 97.2%, with precision and recall values of 93.2% and 92.1%, respectively, for an F-measure of 92.6%. These outstanding performance numbers indicate our approach's ability in properly predicting user attitudes from YouTube comments. We efficiently decreased the dimensionality of the dataset while keeping the most relevant features by including RFE, resulting in enhanced model efficiency and accuracy. The Elastic Net RF with LR approach improved our sentiment classifier even more by managing multicollinearity effectively and boosting predictive power. Furthermore, PCA was critical in improving the model's efficiency by reducing high-dimensional data without compromising crucial information.

Vi. Reference

[1] V. P, R. D and S. N. S. S, "Football Prediction System using Gaussian Naïve Bayes Algorithm," 2023 Second International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2023, pp. 1640-1643, doi: 10.1109/ICEARS56392.2023.10085510.

[2] A.Musdholifah and E. Rinaldi, "FVEC feature and Machine Learning Approach for Indonesian Opinion Mining on YouTube Comments," 2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Malang, Indonesia, 2018, pp. 724-729, doi: 10.1109/EECSI.2018.8752791.

[3] A.N. Muhammad, S. Bukhori and P. Pandunata, "Sentiment Analysis of Positive and Negative of YouTube Comments Using Naïve Bayes – Support Vector Machine (NBSVM) Classifier," 2019

International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), Jember, Indonesia, 2019, pp. 199-205, doi: 10.1109/ICOMITEE.2019.8920923.

[4] Ahmad, I. S., Bakar, A. A., & Yaakub, M. R. (2020). Movie Revenue Prediction Based on Purchase Intention Mining Using YouTube Trailer Reviews. Information Processing & Management, 57(5), 102278. doi:10.1016/j.ipm.2020.102278

[5] Aiyar, S., & Shetty, N. P. (2018). N-Gram Assisted Youtube Spam Comment Detection. Procedia Computer Science, 132, 174–182. doi:10.1016/j.procs.2018.05.181

[1] G. Raju, V. Amudha and S. G, "Comparison of Linear Regression and Logistic Regression Algorithms for Ground Water Level Detection with Improved Accuracy," 2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2023, pp. 1-6, doi: 10.1109/ICONSTEM56934.2023.10142495.

[6] Dabas, C., Kaur, P., Gulati, N., & Tilak, M. (2019). Analysis of Comments on Youtube Videos using Hadoop. 2019 Fifth International Conference on Image Information Processing (ICIIP). doi:10.1109/iciip47207.2019.8985907

[7] Divesh, G. Prasad, G. Sharma and D. K. Vishwakarma, "Sentiment Analysis on Cryptocurrency using Youtube Comments," 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2022, pp. 730-733, doi: 10.1109/ICCMC53470.2022.9753723.

[8] E. Poché, N. Jha, G. Williams, J. Staten, M. Vesper and A. Mahmoud, "Analyzing User Comments on YouTube Coding Tutorial Videos," 2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC), Buenos Aires, Argentina, 2017, pp. 196-206, doi: 10.1109/ICPC.2017.26.

[9] E. Vlachos and Z. -H. Tan, "Public perception of android robots: Indications from an analysis of YouTube comments," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 1255-1260, doi: 10.1109/IROS.2018.8594058.

[10] F. E. Khan, A. M. Ruhan, R. Shamsuddin and F. B. Ashraf, "A Machine Learning Approach to Predict Movie Success from Youtube Trailer Comments," 2022 25th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, 2022, pp. 372-377, doi: 10.1109/ICCIT57492.2022.10055275.

[11] G. Gupta, S. Tiwari, N. Sharma and H. Singh, "Predicting Ratings of Trending Youtube Videos using Machine Learning," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022, pp. 27-31, doi: 10.1109/IC3I56241.2022.10072585.

[12] H. Ding, S. Xie, Y. Xie and J. Wang, "Broad learning system based on Elastic Net feature sparsity and dense," 2021 China Automation Congress (CAC), Beijing, China, 2021, pp. 6646-6650, doi: 10.1109/CAC53003.2021.9727312.

[13] 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.

[14] H. Valpadasu, P. Chakri, P. Harshitha and P. Tarun, "Machine Learning based Spam Comments Detection on YouTube," 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2023, pp. 1234-1239, doi: 10.1109/ICICCS56967.2023.10142608.

[15] K. B. Reddy, M. V. Priya and Murugesan, "Power Theft Detection Using Novel Linear SVM Algorithm and Compared With Convolutional SVM Algorithm For Accuracy," 2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS), Karachi, Pakistan, 2022, pp. 1-9, doi: 10.1109/MACS56771.2022.10022957.

[16] K. L. Sek, P. Lin Lee, K. Y. Pang, Y. Hua, L. Zhu and X. Li, "Principal Component Analysis (PCA) of Surface Contamination by TOF-SIMS," 2021 IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA), Singapore, Singapore, 2021, pp. 1-4, doi: 10.1109/IPFA53173.2021.9617378.

[17] Kavitha, K. M., Shetty, A., Abreo, B., D'Souza, A., & Kondana, A. (2020). Analysis and Classification of User Comments on YouTube Videos. Procedia Computer Science, 177, 593–598. doi:10.1016/j.procs.2020.10.084

[18] Khan, A. U. R., Khan, M., & Khan, M. B. (2016). Naïve Multi-label Classification of YouTube Comments Using Comparative Opinion Mining. Procedia Computer Science, 82, 57–64. doi:10.1016/j.procs.2016.04.009

[19] M. Aufar, R. Andreswari and D. Pramesti, "Sentiment Analysis on Youtube Social Media Using Decision Tree and Random Forest Algorithm: A Case Study," 2020 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 2020, pp. 1-7, doi: 10.1109/ICoDSA50139.2020.9213078.

[20] P. D. Reddy and L. R. Parvathy, "Prediction Analysis using Random Forest Algorithms to Forecast the Air Pollution Level in a Particular Location," 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2022, pp. 1585-1589, doi: 10.1109/ICOSEC54921.2022.9952138.

[21] Pati, B., Panigrahi, C. R., Buyya, R., & Li, K.-C. (Eds.). (2020). Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing. doi:10.1007/978-981-15-1081-6

[22] R. F. Alhujaili and W. M. S. Yafooz, "Sentiment Analysis for Youtube Videos with User Comments: Review," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 2021, pp. 814-820, doi: 10.1109/ICAIS50930.2021.9396049.

[23] R. Pradhan, "Extracting Sentiments from YouTube Comments," 2021 Sixth International Conference on Image Information Processing (ICIIP), Shimla, India, 2021, pp. 1-4, doi: 10.1109/ICIIP53038.2021.9702561.

[24] S. Mitrofanov and E. Semenkin, "An Approach to Training Decision Trees with the Relearning of Nodes," 2021 International Conference on Information Technologies (InfoTech), Varna, Bulgaria, 2021, pp. 1-5, doi: 10.1109/InfoTech52438.2021.9548520.

[25] S. Naveen, S. K. Ravindran, S. G and S. N. Ameen, "Effective Heart disease prediction framework using Random Forest and Logistic regression," 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN), Vellore, India, 2023, pp. 1-6, doi: 10.1109/ViTECoN58111.2023.10157078.

[26] S. Zhou, T. Li and Y. Li, "Recursive Feature Elimination Based Feature Selection in Modulation Classification for MIMO Systems," in Chinese Journal of Electronics, vol. 32, no. 4, pp. 785-792, July 2023, doi: 10.23919/cje.2021.00.347.

[27] W. Wijaya, I. M. Murwantara and A. R. Mitra, "A Simplified Method to Identify the Sarcastic Elements of Bahasa Indonesia in Youtube Comments," 2020 8th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia, 2020, pp. 1-6, doi: 10.1109/ICoICT49345.2020.9166269.

[28] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 2018, pp. 1-2, doi: 10.1109/IWQoS.2018.8624183.