ISSN 2063-5346

# DEEP LEARNING TECHNIQUE FOR SPAM COMMENTS IDENTIFICATION ON YOUTUBE

**A.Yogaraj [1], Snekha D[2], Veena Kumari [2], Tippani Lavanya[3]**

**Article History: Received: 01.02.2023 Revised: 07.03.2023 Accepted: 10.04.2023**

**Abstract**

This paper offers a method for finding spam comments on YouTube, which have been much more prevalent recently. Although YouTube has a spam blocking system in place, it is not always efficient. Then, using information from popular music videos by Psy, Katy Perry, LMFAO, Eminem, and Shakira as comment data, we ran classification experiments using three different deep learning models. Long-Short-Term Memory Networks (LSTM), Longshot-Term Memory Networks (LSTM), and Dense Sequential Classifier Networks (DSCN) (BiLSTM). each model classifies spam comments and non-spam comments. The best model (accuracy high model) is selected and served in real-time applications. YouTube screened spam comments and performed classification experiments using three deep learning models (Dense Sequential Classifier, LSTM, and BiLSTM) that combined these techniques on comment data

***Keywords—component, formatting, style, styling, insert***

[1]Assistant Professor, *Electronics and Communication Engineering. Vel tech High tech Rangarajanm,*
*Dr. Shakuntala engineering college, vel tech High tech Rangarajan Shakuntala, Engineering college*, Chennai, India
[1]Yoga.rajam@yahoo.co.in, [2]snehadhinakaran24@gmail.com, [2]*sveena1001@gmail.com*, [3]tippanilavanya2808@gmail.com

**DOI:10.31838/ecb/2023.12.s1-B.386**

## *I.* INTRODUCTION

The largest video-sharing website in the world, You Tube, was created in 2005 and bought by Google in 2006. The recent transition from internet material to video has led to substantial growth for You Tube as a platform for video content. If spam is any kind of pointless content, then a set of blog comments should be evaluated for them in formativeness by creating a language and tokenization independent metric and determining how much information is present in the text [2]. Users have complete freedom to watch and upload videos. Due to this easy accessibility, there are more personal media outlets now, some of which have grown to be online influencers. To assess and categories the different kinds of comments posted on You Tube by users to show how a two-way

communication tool has been used for both communication and self-expression [3]. 4.5 million videos are seen on You Tube every minute, and there are already over 400 hours of footage there. Users have unrestricted access to watch and post videos. In order to automatically extract and train features with greater generalization power than a collection of words [4], the opinion mining on You Tube comments uses tree kernel technology. Peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement

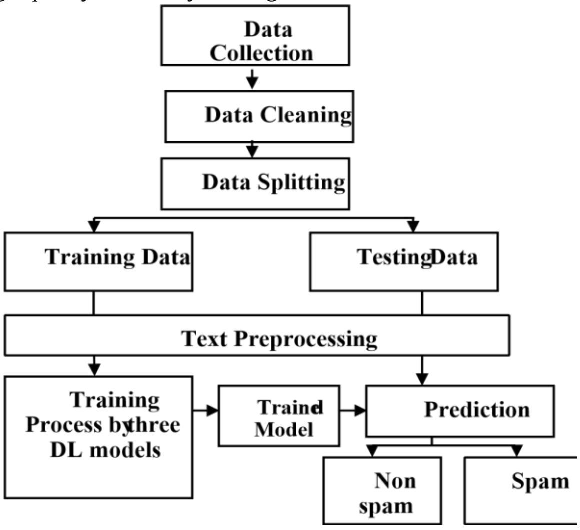
Modelling classifiers are then used to forecast the opinion polarity and the type of remark. Sentiment analysis, commonly referred to as opinion mining, is the study of attitudes and judgements that individuals express in online forums and social media. It also draws viewers to video sharing websites like YouTube. This article provides a succinct overview of methods for analyzing user comments about a given video [5]. A deduction has been made in response to the massive increase in spam comments on YouTube, and classification studies using six different machine learning approaches have been carried out [1]. In order to identify spam comments on YouTube, a deep learning system has been created in this study. By publishing spam in the video comments, spammers can now more easily integrate nefarious activities into websites. In this research, we propose a method for detecting spam comments on YouTube, which have considerably increased in the last few years. On You Tube, there is a spam blocking system in place, but it is not always efficient. conducting classification experiments using three different deep learning models using the data on responses from well-known music videos by Shakira, LMFAO, Eminem, Katy Perry, Psy, etc. Long-Short Term Memory Bidirectional Networks (LSTM), and Dense Sequential Classifier Networks DSCN) (Bi-LSTM). Each model classifies spam comments and non – spam comments. The best model (high accuracy model) is selected and served in Realtime.

## II. RESEARCH METHODS AND MATERIAL

### DETECTING SPAM ON YouTube

A variety of unwelcome threats have recently become present on online social networks. The public is at risk as a result of the misuse of this great resource, even though that they provided us with a free platform to voice our thoughts. For instance, a lot of musicians use YouTube as a platform for promotion, uploading music videos, movie trailers, etc. where viewers can remark on them. However, malicious individuals frequently utilize the comments section to provide links to phishing websites, advertisements, and phone content that may spread malware or viruses. So, it is essential to identify these negative remarks in order to maintain social media's spotless operation. This study used a series of classification methods to distinguish between genuine YouTube video comments and spam. They are effective. The collected 13,000 comments on numerous channels, all of which used the YouTube API to produce music videos, and only considered comments in English. They heuristically categorized the comments by giving each one as Zero as value. To generate accurate findings, The categorization techniques were integrated with Ngam analysis. Their algorithms included Multinomial Naive Bayes, Random Forest, and Support Vector Machine.F1 scores were used to assess how well these models performed. The accuracy ratings for Random Forests and the support vector machine were 0.9726 and 0.9774, respectively [8]. Programmers of all ability levels may watch real developers write, analyze, and run code thanks to video coding tutorials. Past studies in this area have concentrated on explaining the motives and goals of content producers as well as assisting programmers in spotting important content in coding tutorial videos. In this thesis, we emphasize the connection between programmers' coding video viewers. We take a closer look at customer feedback on YouTube coding tutorial videos. Our key goal is to improve content makers' understanding of their audience's needs and preferences so they can respond to these issues more quickly and produce more high-quality content. For our analysis, 6000 comments from 12 YouTube coding videos were utilized in our sample. These days, YouTube receives thousands of submissions per minute, and viewers soon start like and commenting. Millions of comments are posted on some wellknown and popular videos. Some of these comments are positive and helpful, while others are spammy, rude, and occasionally include a URL for commercial advertising or a site redirect. In this study, we employed both standard and artificial neural network-based classifiers to detect spam in datasets of YouTube comments from five well-known singers. The suggested method suggests the most effective classifiers for detecting spam comments by comparing the classifiers' deduced findings [13]. There are now a lot more users on You Tube as a result of its popularity, which has also boosted the number of comments on You Tube channels. We could give

YouTubers tips that would help them create more high-quality content by looking at these




comments. Our study's main objective is the sentiment analysis of comments made in third languages on food related channels. In this study, DBSCAN, an unsupervised learning technique, was used to find distinct patterns in the comment data. Both parametric and nonparametric learning strategies have been modelled and examined. Combining the frequency distribution and logistic regression, 74.01%

1. MODEL OF THE SYSTEM AND SUGGESTED APPROACH

TESTING PROCESS AND ENVIRONMENT

By utilising three deep learning techniques, such as Long Short- Term Memory Networks and Dense Sequential Classifier Networks (LSTM), and LSTM in both directions (Bi-LSTM), our suggested solution is based on identifying spam comments on YouTube. As contrast to conventional learning algorithms, deep learning essentially offers an incredibly fast learning rate and greater generalisation capabilities. The accuracy of the predictions is great. Data cleansing techniques are used after the data has been collected to transform the raw data into processed data. Following that, the data are split into training and testing data for validation. Next we implement text processing to convert text into a numerical value for further processing, which includes tokenization, sequencing, and padding techniques. The three deep learning models mentioned above are then trained using the textpreprocessed training data. Lastly, the trained model used to categorise comments on You Tube is utilised to value the testing data. The python coding used in this paper can be run on any Windows computer (google collab).

2. OVERVIEW FOR THE EXPERIMENT OF THEPROPOSEDTECHNIQUE

As shown in the above we take totally 1983 dataset which then split into training data set (1369%) and testing data set (587%) .TF – IDF vectorization has been used forpreprocessing and six machine learning methods were used and they predict and evaluate the class [1]. A system has been put in place to classify the different kinds of comments individual's post. It seeks to categories the many comment kinds made by YouTube users, demonstrate how to use the interactive feature in a variety of ways, and highlight or categories the purposes of each comment, such as reminiscing, expressing grief, communicating, and giving advice [3]. To ascertain the opinions of the commenters regarding various possibilities Naive Bayes algorithm was used to implement multilabel classification, which required less processing power. According to this reference, no single strategy is effective [7]. As a result, we use many algorithms in our research to discover the optimum categorization approach that produces results with a high degree of accuracy.

3. DATASETS

The datasets used in this paper can be accessed from [16]. It consists basic information about the five most watched music videos from [9]. It includes the following information: You Tube ID, comment author, date, comment content, and labelled classes that describe (0: Ham and 1: Spam). To prevent overfitting, each of these datasets has been pooled into a single collection of data. In this paper we combine all the five datasets of the music videos and by dividing the data into a training dataset (1369) with 70% of the data and a testing dataset (30% of the data) (587)

a) FIGURE 1. Overview of the deep learning process

TABLE 1. Datasets collected and used in the experiments.

| Datasets | Spam | Ham | Total |
|---|---|---|---|
| Psy | 175 | 175 | 350 |
| KatyPerry | 175 | 202 | 350 |
| LMFAO | 236 | 303 | 438 |
| Eminem | 245 | 203 | 448 |
| Shakira | 174 | 196 | 470 |
| Total | 1005 | 978 | 1983 |

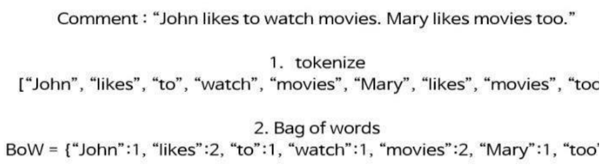



FIGURE 2. Tokenizing and vectorization

4. DATA PROCESSING

Preprocessing is necessary because deep learning models cannot grasp words because the datasets are in texts (convert words into numerical expression). Tokenization is the first stage. The TensorFlow Keras tokenizer API breaks sentences up into words and encodes them into integers by, for example, changing all words to lower case, all words to integer indexes, and then expressing each phrase by a sequence of numbers that will have an identical length. For training and test data, sequencing and padding are performed.

## III. IMPLEMENTATION

1. Data Collection and Cleaning
2. Data Splitting
3. Text Pre-processing
4. Prediction 5. Performance

   evaluation

**1.** Data Collection and Cleaning:

- The data was gathered from a public database.
- The dataset comprises of information from five well-known music videos' comments. It provides the class, comment author, date, and YouTube ID for each remark (0: Ham, 1: Spam). We solely make use of named classes and comment content.
- Overfitting may result from any of the five data sets' training and testing (Table.1) We therefore aggregate all five video datasets in our work in order to generalise the outcome.
- Also, we find missing and redundant data in the original dataset.

**2.** Data Splitting

- A data splitting module has been created to facilitate the analysis of training and test data.
- In this regard, we divide our entire dataset into training and testing data, using 80% of the data for training and 20% for testing.

3. Text Preprocessing

- Text pre-processing, which includes padding, tokenization, and sequencing.

Tokenization - Let's translate text into numerical form since deep learning models cannot comprehend text. Tokenization is a first step in achieving this. Sentences are broken up into words via the TensorFlow Keras Tokenizer API, which then encodes these words as numbers. Tokenizer performs all necessary pre-processing, including

- Filter all punctuation terms, tokenize into words or characters (here we use at the word level)
- Use num words for the maximum amount of unique tokens.
- filter out punctuation terms
- convert all words to lower case .
- tokenize into integer indexes. **4.** Sequencing and Padding: Following tokenization, each sentence is represented by a series of numbers taken from the tokenizer object. Padding is then used to ensure that each sequence is the same length. Both training and test data are subject to sequencing and padding.

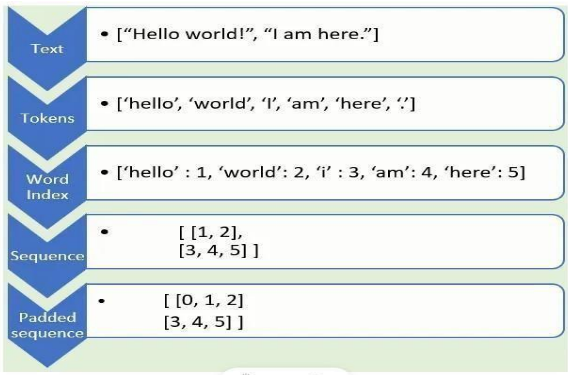



FIGURE 3. Sequencing and Padding

5. Prediction

- After loading and processing our data in the prediction module, we will classify the text message using a neural network design.
- First, train the model using a dense architecture, LSTM, and Bi-LSTM.
- Following that, testing data is validated using trained data.

6. Performance Measure

Accuracy: The analysis of the TP and TN to the total number of test photos is measured.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision: It is an estimating analysis of the ratio of true positives to false positives in the total. It is provided in eqn. (2)

$$Precision = \frac{(TP)}{(TP+FP)} \quad (2)$$

- Recall: In order to calculate the total value of the true positive and false negative rates, the estimate analysis of the true positive rate is used. It is provided in eqn (3).

$$Recall = \frac{(TP)}{(TP+FN)} \quad (3)$$

F-Score: In terms of memory and precision, the FMeasure represents the harmonic mean. The usual F measure balances recall and precision equally (F1).

$$F_{\varepsilon} = (1+\varepsilon^2)\, \frac{Recall \times Precision}{\varepsilon^2 . Precision + Recall} \quad (4)$$

APPLIED DEEP LEARNING ALGORITHMS1) DENSE SEQUENTIAL CLASSIFIER NETWORK

Keras sequential model sequential calls, in which layers are added sequentially.

The embedding layer, the top layer, converts every word into an N-dimensional vector of real values. The size of this vector, which in our case is 16, determines the embedding layer. According to the embedding layer, two words with comparable meanings typically have highly similar vectors. We must pass the form of our input layer as determined by input length (max len = 50) because the embedding layer is the first hidden layer in our model network. The pooling layer assists in preventing over fitting by lowering the number of parameters in the model. Here, we've transformed the layer to one dimension and applied average pooling. Thereafter, in order to avoid over fitting, a dropout layer is applied. This is followed by a dense layer with the activation function "relu," a dense layer with the activation function "sigmoid," and a final output layer. Because there are only two categories to distinguish between (non-spam or spam), we simply use one output neuron. The probability values produced by the sigmoid activation function range from 0 to 1.

7. Long Short-term Memory (LSTM) Model The architecture of LSTM:

To conduct calculations requiring both LSTMs employ the idea of gates to efficiently and effectively combine Short Term Memory (STM) with Long Term Memory (LTM).

1. Forget Gate: LTM deletes useless information when it passes through the forget gate.
2. Learn Gate: STM and Event (current input) are paired to provide, We can apply the essential knowledge we have recently learnt through STM to the current input.
3. Remember Gate: At the Remember gate, the LTM data that we didn't forget, the STM data, and the event data are all combined to produce an updated LTM.
4. Use Gate: As an updated STM, To predict how the present event will turn out, The LTM, STM, and Event gates are also used

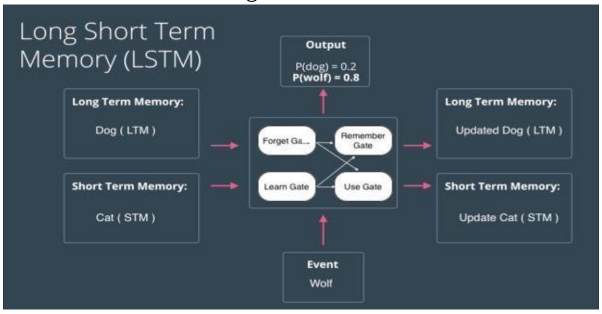



FIGURE 4. Long Short-Term Memory

The preceding figure depicts the streamlined design of the LSTMs. The real mathematical structure of the LSTM is depicted in the picture below.

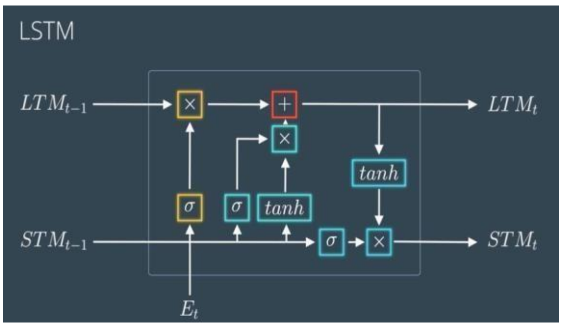


FIGURE 5. Long Short Term Memory

Both short and long-term memory are used by the LSTM described above. Long-term memory has a high data memory sustaining time, making the data difficult to wipe and capable of producing high accuracy evaluation measures.

Bi-LSTM :(Bi-directional long short-term memory):

In essence, bidirectional recurrent neural networks (RNN) are merely two separate RNNs combined. The networks may access both forward and backward information about the sequence because to this structure, which is present at every time step. Bidirectional processing involves processing information in two different directions, one from the present to the future and the other from the future to the present. In contrast to unidirectional methods, this one safeguards data in the backward-running LSTM.
Allowing you to retain both past and present data simultaneously by mixing the two hidden states.

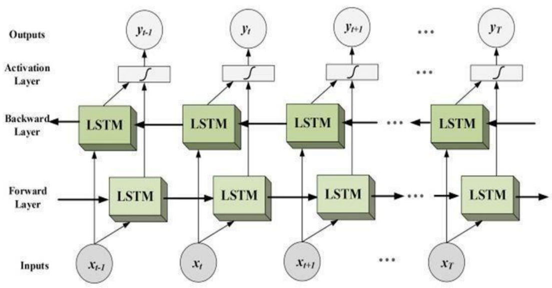



FIGURE 6. Bi-directional Long Short Term Memory

I. RESULT

The below tabular column shows the data set that is combined as a single from the five popular music videos. After combining, we have 1983 comment data sets to be trained and tested.

| index | Comment_Id | Author | date | content | class |
|---|---|---|---|---|---|
| 0 | Z13lggggfffbb3ddzxxtdrd6r6f6ftbyuyguubbvdttf | Dharma pal | 2015- 05 29T02:30:18.971000 | Nice song | 0 |
| 1 | Z13lggggfffbb3ddzxxtdrd6r6f6ftbyuyguubbvdttf | Tiza arellano | 2015- 05 29T00:30:18.971000 | I love song | 0 |
| 2 | Z13lggggfffbb3ddzxxtdrd6r6f6ftbyuyguubbvdttf | Princessalis leve demine madis | 2015- 05 28T21:30:18.971000 | I love song | 0 |
| 3 | Z13lggggfffbb3ddzxxtdrd6r6f6ftbyuyguubbvdttf | Eric gonzalez | 2015- 05 29T02:30:18.971000 | 860,00,0000 lets make it first female to reach one billion! Share it and replay it! | 0 |
| 4 | Z13lggggfffbb3ddzxxtdrd6r6f6ftbyuyguubbvdttf | Analena lopez | 2015- 05 29T02:30:18.971000 | Shakira is best for worldcup | 0 |
| 5 | Z13lggggfffbb3ddzxxtdrd6r6f6ftbyuyguubbvdttf | Jehoiada wellington | 2015- 05 29T02:30:18.971000 | The best world cup song ever!! | 0 |
| 6 | Z13lggggfffbb3ddzxxtdrd6r6f6ftbyuyguubbvdttf | Kara cuthbertson | 2015- 05 29T02:30:18.971000 | I love | 0 |
| 7 | Z13lggggfffbb3ddzxxtdrd6r6f6ftbyuyguubbvdttf | Sudheer yadav | 2015- 05 29T02:30:18.971000 | SEE SOME MORE SONG OPEN GOOGLE AND TYPE shakira guruvof movie | 1 |
| 8 | Z13lggggfffbb3ddzxxtdrd6r6f6ftbyuyguubbvdttf | Alex john | 2015- 05 29T02:30:18.971000 | Awesome | 0 |
| 9 | Z13lggggfffbb3ddzxxtdrd6r6f6ftbyuyguubbvdttf | Nirab valobasha | 2015- 05 29T02:30:18.971000 | I like shakira | 0 |
| 10 | Z13lggggfffbb3ddzxxtdrd6r6f6ftbyuyguubbvdttf | Chelsea andrews | 2015- 05 29T02:30:18.971000 | Shakira waka waka <br/> LOVE THIS SONG!!!!!!!!! | 0 |

Dataset number of rows and cols/ data types, memory usage (1956,5)

<class ‘pandas.core.frame.DataFrame’>
Int64Index: 1956 entries, 0 to 349
Data columns (total 5 columns):

| # | column | non-null | count | Dtype |
|---|---|---|---|---|
| 0 | Comment Id | 1956 | Non-Null | Object |
| 1 | Aurthor | 1956 | Non-Null | Object |
| 2 | Date | 1711 | Non-Null | Object |
| 3 | Content | 1956 | Non-Null | Object |
| 4 | Class | 1956 | Non-Null | Object |

D types: int64(1),object(4)
Memory usage:91.7+kb
None

DATA CLEANING:

| INDEX | CONTENT | CLASS |
|---|---|---|
| 0 | Nice song | 0 |
| 1 | I love song | 0 |
| 2 | I love song | 0 |
| 3 | 860,00,0000 lets make it first female to reach one billion! Share it and replay it! | 0 |
| 4 | Shakira is best for world cup | 0 |
| 5 | The best world cup song ever!! | 0 |
| 6 | I love | 0 |
| 7 | SEE SOME MORE SONG OPEN GOOGLE AND TYPE shakira guruvof movie | 1 |
| 8 | Awesome | 0 |

| | | |
|---|---|---|
| 9 | I like shakira | 0 |
| 10 | Shakira -waka waka <br/> LOVE THIS SONG!!!!!!!!! | 0 |
| 11 | Why so many disliked????!!!!! | 0 |
| 12 | I don' t think this song will ever get old | 0 |
| 13 | Love song | 0 |
| 14 | Wary good | 0 |
| 15 | Every time I hear this song, I think about Iniesta' s goal against the Netherlands | 0 |
| 16 | Whose watching this in 2015. If so hi-5 | 0 |

From the attributes of comment Id, comment author, date ,comment content and labeled classes. We need only comment content and class. So data cleaning is made to remove the unwanted data and segregate the needed data.

Null Checking result

Content: 0
Class : 0
Ditype : int64

Null checking result in checking if there is any missing data or null columns. As from the above we don't have any null columns so it resulted as 0.

MODULE 2 – Data Splitting

Data Splitting

Shape of x_train is (1564,)
Shape of x_test is (392,)

Data splitting is nothing but splitting the datasets into training data set and testing data set. After training the model, test data sets have to be given to label the inputs whether it is spam or ham.

MODULE 3 – Text Preprocessing

Tokenization and Word Index Result

Tokenization and word Index Result

'epic': 823,

'mix': 824,

'passed': 825,

'ha': 826,

'place': 827,

104999962146104962510':

828,

'robot': 829,

'movie': 830,

'sure':831,

'message': 832,

'said':833,

'damn': 834,

'clip': 835,

'subscribe': 836,

The above sample shows the comments that are tokenized (separating the words) and indexing has been done for each comment.

Sequencing and padding for training data

Sequencing and padding – Testing Data
[[4,23,15,303,188,1,106,1],
[4,5,68,23,2,152,1,19,3],
[107,2,180,62,1,4,2,1,5,23,100,153,1,3],[67,46,6,352,
10,8,1,1,20,1]

[4 23 15 …….. 0 0 0]
[4 5 68 ……. 0
0 0] [107 2 180 …….. 0 0 0] …..
[10 8 4 ……0 0
0] [71 15 9…….
0 0 0]
[1 3 0……..0 0 0]]

The training data has been sequenced and padding has been done to equal the lengths.

Sequencing and padding for testing data

Sequencing and padding – Testing Data

[[234, 61, 10, 12, 32, 6, 27, 326, 47, 420, 8, 7, 54, 14, 3], [2, 31, 5, 140, 11,1], [10, 8, 12, , 40, 1, 259, 17, 1,
26, 42, 107, 81, 54,

[[234 61 10……. .0 0 0]
[ 2 31 5………0
0 0] [204 1
0………0 0 0]
……
[11 46 127………1 5 1]
[ 2 1 1………
0 0 0] [70 15

141……...0 0

0]

The testing data has been sequenced and padding has been done to equal the lengths.

MODULE 4 – Prediction

Dense Sequential Classifier Training Process

Epoch1/30:49/49 65 loss: 8.6898 accuracy: 0.5345 val loss:
8.6845 val accuracy: 0.5689 - 6s/epoch - 119ms/step
Epoch 2/30:49/49-2s - loss: 8.6487 accuracy: 8.6324 - val loss:
8.6012 - val_accuracy: 8.6556 25/epoch - 35ms/step
Epoch 3/30:49/49-25- loss: 0.4158 accuracy: 0.8485 val loss:
0.3265 val_accuracy: 0.9831 25/epoch 38ms/step
Epoch 4/30:49/49 25 1055: 0.2561 accuracy: 8.9290 val loss:
8.3113 val_accuracy: 0.9885 25/epoch 35es/step
Epoch 5/36:49/49-15 loss: 0.2573- accuracy: 8.9214 val loss: 0.4026 Val accuracy: 8.8903 - 15/epoch - 27ms/step Epoch 6/30:49/49 15 loss: 0.2613 accuracy: 8.9220 val loss: 6.2697-val accuracy: 0.9158 15/epoch - 26ms/step
Epoch 7/30:49/49-15 loss: 8.2163 accuracy: 0.9373 val loss:
0.2766 - val_accuracy: 8.9209 - 1s/epoch - 25ms/step -
Epoch 8/30:49/49-15 - loss: 0.1754 accuracy: 0.9588 val loss:
8.2207 val_accuracy: 0.9286 1s/epoch- 25ms/step
Epoch 9/30:49/49 is loss: 0.1458- accuracy: 8.9636 val loss:
0.2893-val accuracy: 8.9184 1s/epoch- 25ms/step
Epoch 10/30:49/49-15 loss: 8.1362 - accuracy: 8.9636 val loss:
8.2248 - val_accuracy: 0.93111s/epoch – 25ms/step

In this paper, the first algorithm is DSCN, after training this model testing data has been given to this model and here we give 30 epochs in order get a proper accuracy , so that each single comment dataset iterates till 30 times until it gets a constant accuracy.

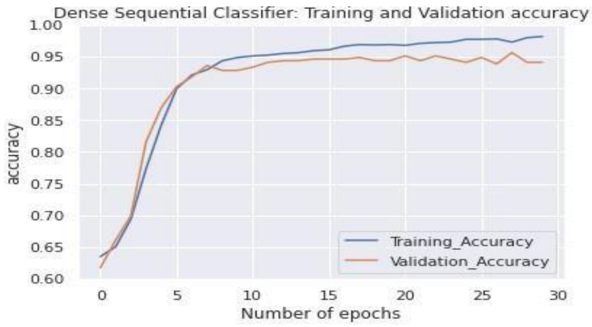



FIGURE 7. DSC: Training and Validation accuracy The above graph shows the training accuracy and the testing accuracy

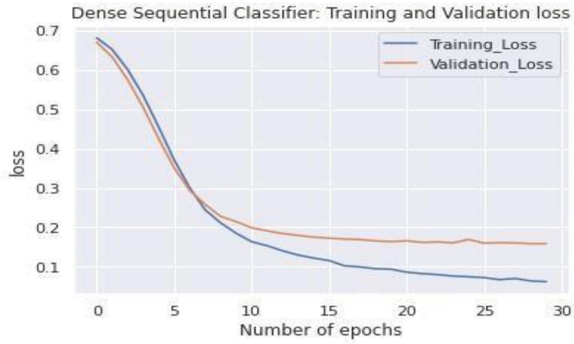



FIGURE 8. DSC: Training and Validation Loss

The above graph shows the training loss and the testing loss .

The predicted accuracy is then given to the trained model and a confusion matrix has been generated. The diagonal part shows the predicted and spam and ham comments. and the dark one gives the validation losses.

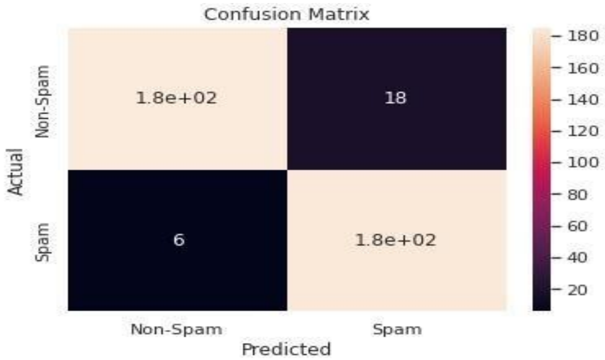



FIGURE 9. Confusion Matrix

Accuracy: [0.16617940366268158, 0.956632673740387]

This shows the accuracy of training and testing dataset
from DSCN model

Testing Process - Prediction LSTM Training Process

| | | | |
|---|---|---|---|
| Accuracy 392 | | | 0.93 |
| macro avg 392 | 0.93 | 0.93 | 0.93 |
| weighted avg 392 | 0.93 | 0.93 | 0.93 |

Epoch 1/30:49/49 - 6s - loss: 8.6898 - accuracy: 0.5345 - val loss: 0.6845 - val accuracy: 8.5689 - 6s/epoch- 119ms/step
Epoch 2/30:49/49 - 25 - loss: 0.6487 - accuracy: 0.6324 - val loss: 0.6012 Val accuracy: 0.6556 - 2s/epoch- 35ms/step
Epoch 3/30:49/49 - 2s - loss: 8.4158 - accuracy: 0.8485 val loss: 0.3265 - val_accuracy: 0.9031 - 2s/epoch - 38ms/step
Epoch 4/30:49/49-2s - loss: 0.2561 accuracy: 0.9290 val loss: 0.3113 - val_accuracy: 0.9005 25/epoch - 35ms/step
Epoch 5/30:49/49 - 1s - loss: 0.2573- accuracy: 0.9214 val loss: 0.4026 val_accuracy: 0.8903- 1s/epoch - 27ms/step
Epoch 6/30:49/491s loss: 0.2613 - accuracy: 0.9220 val loss: 0.2697-val accuracy: 0.9158 1s/epoch 26ms/step
Epoch 7/38:49/49 is-loss: 8.2163 accuracy: 0.9373 val loss: 0.2766 - val accuracy: 0.9209 - 1s/epoch - 25ms/step
Epoch 8/30:49/49 is-loss: 8.1754 accuracy: 0.9508 val loss: 0.2207 - val_accuracy: 0.9286 1s/epoch- 25ms/step
Epoch 9/30:49/49-1s- loss: 8.1458- accuracy: 0.9636 val loss: 0.2893 val accuracy: 8.9184 1s/epoch - 25ms/step
Epoch 10/38:49/49-is-loss: 8.1362 - accuracy: 0.9636 val loss: 0.2240 val accuracy: 0.93111s/epoch - 26ms/step

In this paper, the second algorithm is LSTM, after training this model testing data has been given to this model and here we give 30 epochs in order get a proper accuracy , so that each single comment dataset iterates till 30 times until it gets a constant accuracy.

FIGURE 10. LSTM- Training and Validation accuracy

The above graph shows the training accuracy and the testing accuracy.

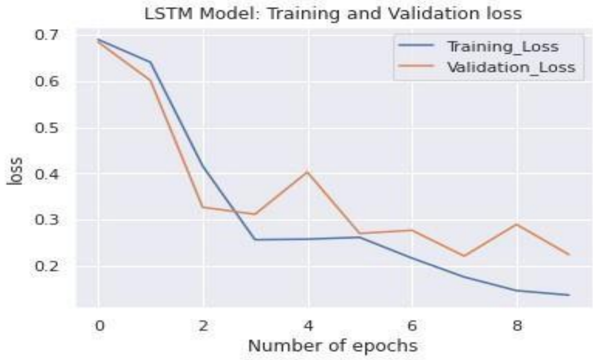



FIGURE 11. LSTM- Training and Validation accuracy

The above graph shows the training loss and the testing loss.

The predicted accuracy is then given to the trained model and a confusion matrix has been generated. The diagonal part shows the predicted and spam and ham comments. and ne gives the validation losses.

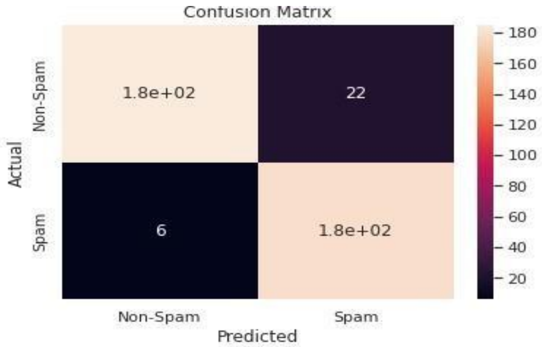



FIGURE 12. Confusion Matrix

Accuracy: [0.22403298318386078, 0.9311224222183228

| score support | precision | recall | f1- |
|---|---|---|---|
| 0 191 | 0.89 | 0.97 | 0.93 |
| 1 201 | 0.97 | 0.89 | 0.93 |

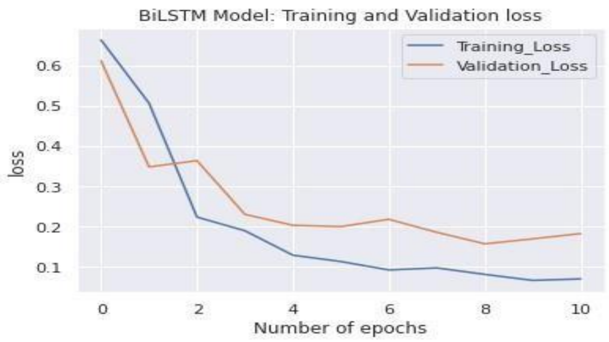
This shows the accuracy of training and testing dataset from LSTM model.

Bi-LSTM

Training Process Epoch 1/30:49/49-18s loss: 0.6635 accuracy: 0.6528 val loss: 0.6119 Val accuracy: 88 18s/epoch -196ms/step

Epoch 2/38:49/49-3s- loss: 8.5068 - accuracy: 0.7685 val loss:
0.3487 Val accuracy: 0.8571 3s/epoch-
70ms/step
Epoch 3/38:49/49-2s loss: 0.2241 accuracy: 0.9194 val loss:
0.3641 Val accuracy: 0.8061 2s/epoch -
38ms/step
Epoch 4/38:49/49-25- loss: 0.1901 - accuracy: 0.9322 val loss:
0.2307 Val accuracy: 0.9158 25/epoch
37ms/step
Epoch 5/30:49/49-25 loss: 0.1295 accuracy: 0.9546 val loss:
0.2038 Val accuracy: 0.9235 2s/epoch
37ms/step
Epoch 6/30:49/49-25- loss: 0.1138 accuracy: 8.9559 - Val loss:
8.2005 Val accuracy: 0.9311 - 2s/epoch - -
- 37ms/step
Epoch 7/30:49/49-25 - loss: 0.8929 - accuracy: 8.9674 Val loss:
0.2186 Val accuracy: 0.9107 - 2s/epoch-
37ms/step
Epoch 8/38:49/49-35-loss: 8.8979 accuracy: 0.9674 Val loss: 0.1862 - Val accuracy: 0.9260 3s/epoch -
52ms/step
Epoch 9/38:49/49-35-loss: 8.8819 accuracy: 8.9757 Val loss:
0.1577 Val accuracy: 0.9439 - 3s/epoch-
65ms/step
Epoch 18/38:49/49-25 loss: 8.8669- accuracy: 0.9815 Val loss:
0.1700 Val accuracy: 0.9464 25/epoch -
37ms/step
Epoch 11/30:49/49-25 loss: 8.8786- accuracy: 8.9763 Val loss:
0.1832 Val accuracy: 0.9464 25/epoch - 38ms/step

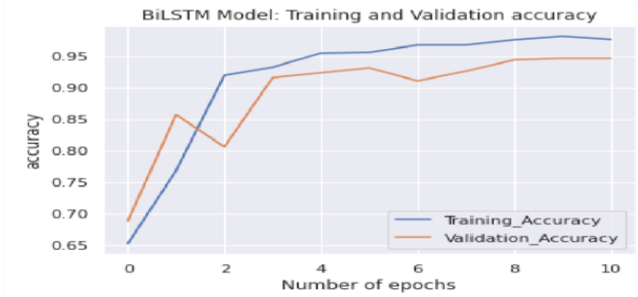
In this paper, the second algorithm is LSTM, after




training this model testing data has been given to this model and here we give 30 epochs in order get a proper accuracy , so that each single comment dataset iterates till 30 times until it get a constant accuracy.

BiLSTM: Training and Validation Loss

The above graph shows the training accuracy and the testing accuracy.

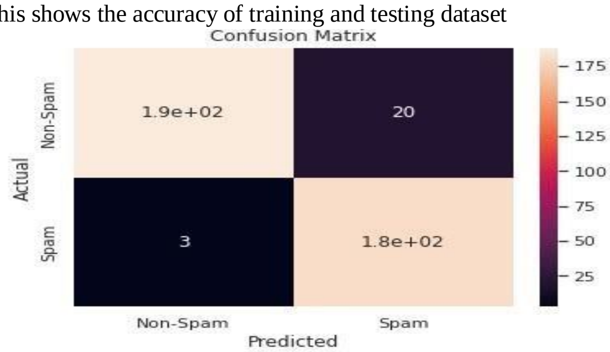



BiLSTM: Training and Validation

After comparing the accuracy of all the three models and a single data test prediction is done with that well

I. Testing Process

This shows the accuracy of training and testing dataset




## 5. CONCLUSION:

In this paper, we proposed a method for detecting spam comments on YouTube, which has been rapidly increasing in recent years, using a deep learning model. YouTube screened spam comments and performed classification experiments using three deep learning models (Dense Sequential Classifier, LSTM, and LSTM) that combined data. The effectiveness of the suggested model was demonstrated through the results of quantitative evaluation metrics such as accuracy, f1 score, recall showed that Dense Sequential Classifier Networks proposed in this paper is the most precise.

## 6. VI. REFERENCES:

A. Aziz, C. F. M. Foozy, P. Shamala, and Z. Suradi, "Youtube spam comment detection using support vector machine and k-nearest neighbor," Tech. Rep., 2018, doi: 10.11591/ijeecs.v12.i2.pp607-611

[2] R. K. Das, S. S. Dash, K. Das, and M. Panda,

"Detection of spam in Youtube comments using different classifiers," in Advanced Computing and Intelligent Engineering, 2020, pp. 201–214, doi: 10.1007%2F978- 981-15-1081-6_17.

[3] N. M. Samsudin, C. F. B. Mohd Foozy, N. Alias, P. Shamala, N. F. Othman, and W. I. S. Wan Din, "Youtube spam detection framework using Naïve Bayes and logistic regression," Indonesian J. Electr. Eng. Comput. Sci., vol. 14, no. 3, p. 1508, Jun. 2019,doi: 10.11591/ijeecs.v14.i3.pp1508-1517.

[4] G. Kaur, A. Kaushik, and S. Sharma, "Cooking is creating emotion: A study on hinglish sentiments of Youtube cookery channels using semisupervised approach," Big Data Cognit. Comput., vol. 3, no. 3, p. 37, Jul. 2019, doi: 10.3390/bdcc3030037.

[5] E. Ezpeleta, M. Iturbe, I. Garitano, I. V. de Mendizabal, and U. Zurutuza, "A mood analysis on Youtube comments and a methodfor improved social spam detection," in Proc. HAIS, 2018, pp. 514–525, doi: 10.1007/978-3- 319-92639-1_43.

[6] N. Hussain, H. Turab Mirza, G. Rasool, I. Hussain, and M. Kaleem, "Predilection decoded: Spam review detection techniques: A systematic literature review," Appl. Sci., vol. 9, no. 5, p. 987, Mar. 2019, doi: 10.3390/app9050987.

[7] L. Song, R. Y. K. Lau, R. C.-W. Kwok, K. Murkowski, and W. Dou, "Who are the spoilers in social media marketing? Incremental learning of latent semantics for social spam detection," Electron. Commerce Res., vol. 17, no. 1, pp. 51–81, Mar. 2017, doi: 10.1007/s10660-016-9244-5.

[8] S. Jain and D. M. Patel. (2019). Analyzing User Comments of Learning Videos From YouTube Using Machine Learning. [Online]. Available:

[9] P. Bansal. (2019). Detection of Offensive YouTube Comments, a Performance Comparison of Deep Learning Approaches. [Online]. Available: https://core.ac.uk/reader/30131303

[10]G. Shi, F. Luo, Y. Tang, and Y. Li, 'Dimensionality reduction of hyperspectral image based on local constrained manifold structure collaborative preserving embedding," Remote Sens., vol. 13, no. 7, p. 1363, Apr. 2021, doi: 10.3390/rs13071363.