YOUTUBE SPAM DETECTION USING SUPPORT VECTOR MACHINE (SVM) ALGORITHM

A Project Report Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

Submitted By

P Sisindri	18221A0576
P M Praveen	18221A0578
P R K Sai Prasad	18221A0580
T N Lokesh	19225A0511

Under the Supervision of

Mr B. Srinivas

M.Tech Assoc Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by A.I.C.T.E, New Delhi & Permanently Affiliated to J. N.T.U.K, Kakinada)
(Accredited by N.B.A & NAAC with 'A' Grade) **ODALAREVU–533210**

2018-22

BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by A.I.C.T.E, New Delhi & Permanently Affiliated to J. N.T.U.K, Kakinada)

(Accredited by N.B.A & NAAC with 'A' Grade)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **" YOUTUBE SPAM DETECTION USING SUPPORT VECTOR MACHINE (SVM) ALGORITHM "** is being submitted for the partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering**, at **BVC Engineering College**, **Odalarevu**, is a bonafide work done by **P . Sisindri (18221A0576), P .M. Praveen (18221A0578) P. R. K . Sai Prasad (18221A0580), T. N .Lokesh (19225A0511),** under the academic year 2021-22 and it has been found suitable for acceptance according to the requirement of University. The results embodied in this thesis have not been submitted to any other University Institute for the award of any degree.

Project Guide

Mr B.Srinivas
Associate Professor

Head of the Department

Dr. G Jena
M.Tech. Ph.D
Professor

External Examiner

ACKNOWLEDGEMENT

We express our deep sense of gratitude and respect towards our guide **Mr B. Srinivas AssocProfessor** of CSE Department, BVC Engineering College, Odalarevu for his excellent guidance right from selection of the Project work. Her constant encouragement and support have been the cause of our success in completing this thesis in the college.

We are thankful To **Dr. GUNAMANI JENA**, **Professor and Head of the Department of CSE** for facilitating, Training, providing us the counselling and support to carry out the project work.

We are also thankful to **Dr. Chandramouli V S A**, **Principal**, BVC Engineering College, Odalarevu, for his support during and till the completion of the project.

We are thankful to all the **teaching and non-teaching staff** of Computer Science & Engineering Department, **Management**, BVC Engineering College, Odalarevu, and our friends for their direct and indirect help provided to us in completing the project.

Project Associates:

P. Sisindri	**18221A0576**
P. M. Praveen	**18221A0578**
P. R. K. Sai Prasad	**18221A0580**
T. N. Lokesh	**19225A0511**

CONTENTS

ABSTRACT

SPAMMING is the one of messaging system to send an unsolicited message. YouTube is one of the biggest sites for the user to get information. The best thing about the YouTube is the user can subscribe the channel, like or dislike the video and also giving opinion on the comment section on that video, and YouTube has attracted the number of users. This attracts the spammers by spamming the comments. The spam comments on YouTube offers limited tools for comment moderation, so that the spam volume is shockingly increased which leads to owners disables the comments.

Thus industries and researchers have applied completely different approaches to form spam free social network platforms. The survey for the spam comments prediction is done by using different **Machine Learning** algorithms.

INTRODUCTION

In the previous years, informal online communities like Facebook and YouTube have become progressively common platform in an individual person's day to day life. People use social media as a virtual community platform to stay in touch with friends and family and to also share thoughts and ideas in blogs. Due to this developing pattern, these platforms pull in an enormous number of clients and are easy targets for spammers. Youtube has become the most well – known informal community among youngsters. These days, 200 million clients produce 400 million new YouTube content i.e.,videoseveryday. This extensive environment provided by YouTube also creates an opportunity for spammers to create irrelevant content directed to users. These irrelevant or unsolicited messages are aimed to attack users by luring them into clicking links to view malicious sites containing malware and scams.

One of the most highlighted features of YouTube is the comments section below every video posted by a user. This feature allows users to share opinions and ideas.

In this project, the prediction of the spam comments present in the comments section of YouTube videos using the concept called machine learning, it is also known as subset of Artificial Intelligence.

1.1 Spam Detection Approach

YouTube is not excluded from malicious user who are often found to expose in spamming and promotional activities. There are many approaches to detect Spam such as using Artificial Intelligent, Cryptography, Machine Learning and others. However, Manwar said the machine learning also capable to detect YouTube spam. The existing study in YouTube Spam Detection is Manwar and Alberto show that both of the authors used Support Vector Machine (SVM) as a classifier in classification phase. Manwar stated that SVM classification is in binary-two class. Usually, class denoted by 0 and 1. However, the collection data have been classified into two classes. Hence, easy for pre-processing and feature selection to perform.

1.2 Online Social Media Sites

In the recently advanced society, online social media sites like YouTube, Twitter, Facebook, LinkedIn, etc are very popular. People turn to social media for interacting with other people, gaining knowledge, sharing ideas, for entertainment and staying informed about the events happening in the rest of the world. Among these sites, YouTube has emerged as the most popular website for sharing and viewing video content. However, such success has also attracted malicious users, which aim to self-promote their videos or disseminate viruses and malware. These spam videos may be unrelated to their title or may contain pornographic content. Therefore, it is very important to find a way to detect these videos and report them. In this work, we have evaluated several top-performance classification techniques for such purpose. The statistical analysis of results indicates that the Multilayer Perceptron and Support Vector Machine show good accuracy results.

1.3 Motivation:

According to a press release by Google, more than a million advertisers are using Google ad platforms, the mobile profits on YouTube are up 100% year over year and the number of hours people are watching on YouTube each month is up 50% year over year. At the same time, according to Nexgate, a computer security company, just in the first half of 2013, the volume of social spam increased by 55%. For each spam found on any social network, the other 200 spams are found on Facebook and YouTube. The problem became so dangerous that it motivated users to create a request in 2012, in which they ask YouTube to provide tools to deal with undesired content. In 2013, the YouTube official blog report efforts to deal with undesired remarks through recognition of malevolent links, ASCII art detection and display changes to long comments. However, many users are still not satisfied with such solutions. In fact, in 2014, the user “PewDiePie”, owner of the most subscribed channel on YouTube (nearly 40 million subscribers), disabled comments on his videos, claiming most of the comments are mainly spam and there is no tool to deal with them.

1.4 Problem Statement:

We use Artificial Neural Networks (ANN) to predict the comments Spam or Ham (Not Spam).

1.5 Objective:

Here, we are applying different Machine Learning algorithms such as Logistic Regression, Support Vector Machine (SVM) and Decision Tree to predict the comments whether they are Spam or Ham and compare these algorithms with each other to find which algorithm is more accurate.

1.6 Scope of work:

Artificial Neural Networks(ANN) is applied to data to identify the comments Spam or Ham. Again we are using different Machine Learning algorithms such as Logistic Regression, Support Vector Machine(SVM) and Decision Tree to predict the comments whether they are Spam or Ham.

1.7 Applications:

These applications are helpful Music apps

1.8 Organization of Report:

The rest of the report is organized into 5 chapters. After this introductory chapter, the next chapter-2 describes about the survey of the existing system. This establishes a context of the research conducted by the researchers up until now in the field of Spam Detection using ML and DL algorithms.

Chapter-3 describes the proposed system. This starts with the introduction of the dataset, the models that have been used in the report. Then it covers the architecture of the proposed system. Describes the process and the algorithms used, the details of the software used for the research work. It also describes the evaluation parameters used for this study.

Chapter-4 shows the experiment and the results. It appears the confusion network of each model and the comparison graph. This helps us to identify which model is the most efficient for the stock market trend prediction using ML and DL algorithms. .

Chapter-5 gives a conclusion about the result of all the models in this research paper and gives suggestions about which model to use when. It gives a new direction of future work.

2 Literature Survey

1. YouTube spam comment detection using support vector machine and K–nearest neighbour:

Social networking such as YouTube, Facebook and others are very popular nowadays. The best thing about YouTube is user can subscribe also giving opinion on the comment section. However, this attract the spammer by spamming the comments on that videos. Thus, this study develop a YouTube detection framework by using Support Vector Machine (SVM) and K-Nearest Neighbor (k-NN). There are five (5) phases involved in this research such as Data Collection, Pre-processing, Feature Selection, Classification and Detection. The experiments is done by using Weka and RapidMiner. The accuracy result of SVM and KNN by using both machine learning tools show good accuracy result. Others solution to avoid spam attack is trying not to click the link on comments to avoid any problems.

2. A Survey on Spam Detection Methodologies in Social Networking Sites

Conventional media, such as television or newspapers, essentially transmits information in one direction. Social media is a two-way form of communication that allows users to interact with the information being transmitted. Social media encompasses a wide variety of online content, from social networking sites like Facebook .online social networks are becoming popular among internet users. The internet users spend more amount of time on popular networking sites like Facebook, Twitter, google+ etc. Huge information available on these sites attracts the spammers who misuse the valuable information on these sites. Spammers send unwanted messages, share malicious links, develop malicious apps and sometimes create fake accounts. A lot of research has been done to detect spam on social networking sites. In this paper we have reviewed different research papers on spam detection. Our study provides techniques used, dataset and accuracy of various spam detection methodologies.

3. KidsTube: Detection, Characterization and Analysis of Child Unsafe Content & Promoters on YouTube

YouTube draws large number of users who contribute actively by uploading videos or commenting on existing videos. However, being a crowd sourced and large content pushed onto it, there is limited control over the content. This makes malicious users push content (videos and comments) which is inappropriate (unsafe), particularly when such content is placed around cartoon videos which are typically watched by kids. In this paper, we focus on presence of unsafe content for children and users who promote it. For detection of child unsafe content and its promoters, we perform two approaches, one based on supervised classification which uses an extensive set of video-level, user-level and comment-level features and another based Convolutional Neural Network using video frames. Detection accuracy of 85.7% is achieved which can be leveraged to build a system to provide a safe YouTube experience for kids. Through detailed characterization studies, we are able to successfully conclude that unsafe content promoters are less popular and engage less as compared with other users. Finally, using a network of unsafe content promoters and other users based on their engagements (likes, subscription and playlist addition) and other factors, we find that unsafe content is present very close to safe content and unsafe content promoters form very close knit communities with other users, thereby further increasing the likelihood of a child getting getting exposed to unsafe content.

4. How Useful Are Your Comments? Analyzing and Predicting YouTube Comments and Comment Ratings

An analysis of the social video sharing platform YouTube reveals a high amount of community feedback through comments for published videos as well as through meta ratings for these comments. In this paper, we present an in-depth study of commenting and comment rating behavior on a sample of more than 6 million comments on 67,000 YouTube videos for which we analyzed dependencies between comments, views, comment ratings and topic categories. In addition, we studied the influence of sentiment expressed in comments on the ratings for these comments using the SentiWordNet thesaurus, a lexical WordNet-based resource containing sentiment annotations. Finally, to predict community acceptance for comments not yet rated, we built different classifiers for the estimation of ratings for these comments. The results of our large-scale evaluations are promising and indicate that community feedback on already rated comments can help to filter new unrated comments or suggest particularly useful but still unrated comments.

5. Classification Methods for Spam Detection Online Social Network

In the recent advanced society the online social networking sites like Twitter, Facebook, LinkedIn are very popular. Twitter, an online Social Networking site, is one of the most visited sites. Lot of users communicates with each other using Twitter. The rapidly growing social network Twitter has been infiltrated by large amount of spam. As Twitter spam is not similar to traditional spam, such as email and blog spam, conventional spam filtering methods are not appropriate and effective to detect it. Thus, many researchers have proposed schemes to detect spammers in Twitter, so need to identify spammers in twitter. Spam detection prototype system is proposed to identify suspicious users and tweets on Twitter. The proposed approach is to identify spam in Twitter using template, content, user based features to analyze behavior of user. Twitter API is used to get all details of twitter user and then generate the template. This template generated is then matched with predefined template. If suspicious behavior is analyzed, the account is considered as spam. However in case spam is not detected, the system collects 'content based' and 'user based' features from twitter account, by using the 'feature matching technique' to match features. Algorithms used in the proposed system are supported by machine learning, which is used to match features and identify spam. Two Classification Algorithms, Naive Bayes and Support Vector Machine, are used for providing better

accuracy and reducing execution time by the use of Template Matching. Public Dataset is collected from internet for providing training to Naive Bayes and Support Vector Machine classifiers.

2.2 Research Contribution:

Research has been conducted for spam detection author used only SM algorithm but we have added decision tree and logistic regression and ANN algorithm and then calculating accuracy, precision, recall and FSCORE between all algorithms and this application will split dataset into train and test where 80% dataset will be used for training and 20% will be used for testing and this dataset split is based on random so accuracy may vary during each execution.

3 Proposed System

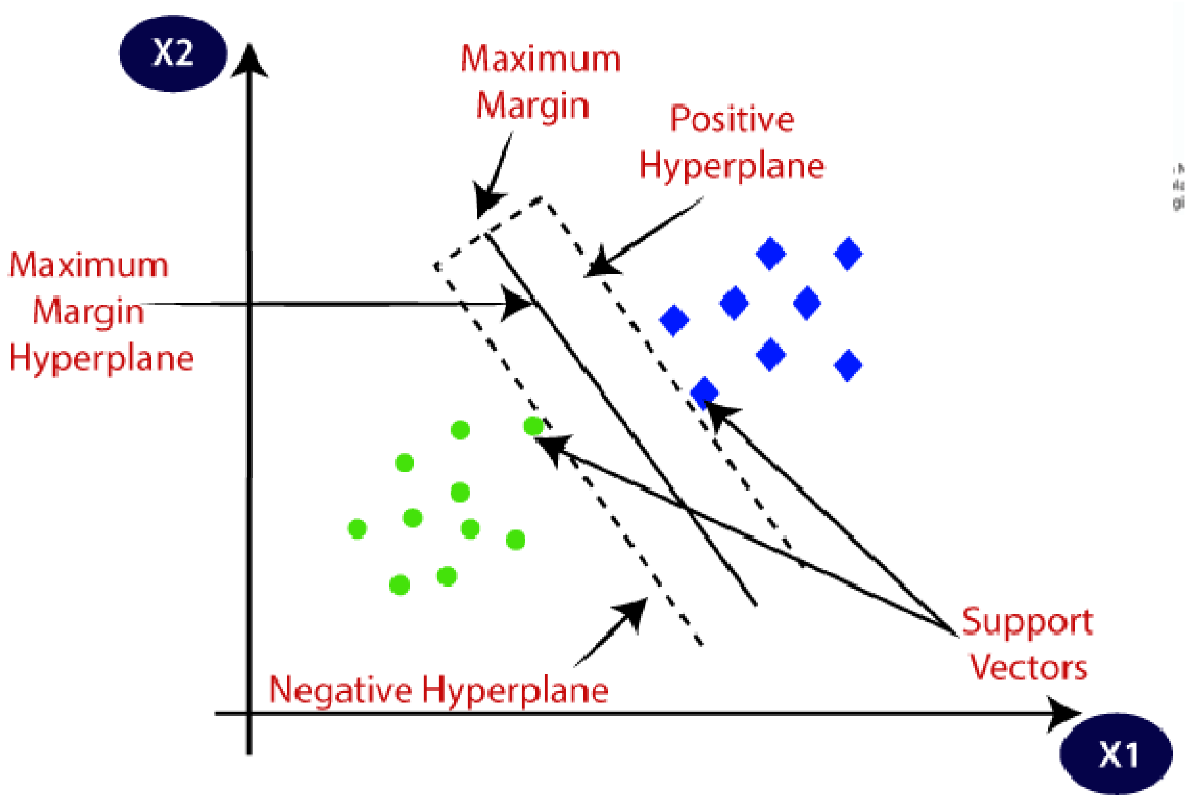
In proposed system we are applying different Machine Learning algorithms such as Logistic regression, Support vector machine(SVM) and Decision Tree to predict the comments whether they are SPAM or HAM and compare these algorithms with ANN to find which algorithm is more accurate.

3.1 Algorithms:

In this study, we use nine machine learning methods (Decision Tree, SVM, and two deep learning algorithms (Feed forward neural network and LSTM).

3.1.1 SVM(SUPPORT VECTOR MACHINE):

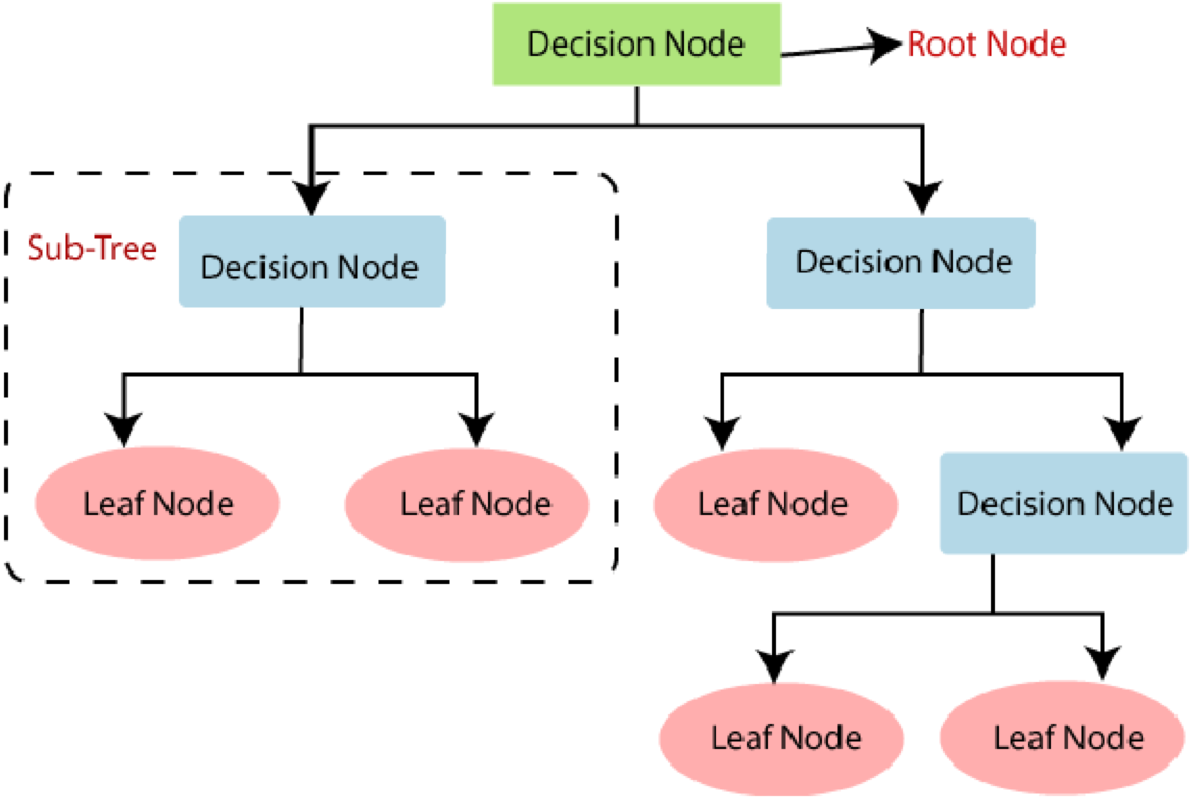
Support Vector Machine (SVM) is a supervised machine learning algorithm. This model has better prediction performance in short and medium term compared to long term. Every algorithm has its way of learning patterns and then predicting. The predictability of financial trend with SVM model by evaluating the weekly trend of NIKKEI 225 index. SVM is a boundary that best separates two classes with employing a line or hyperplane. The decision boundary is defined in Equation. SVMs convert non-separable classes to separable ones by kernel functions such as linear, non-linear, sigmoid, radial basis function (RBF) and polynomial.



Fig(I) Support Vector Machine

3.1.2 Decision Tree Algorithm:

The purpose is to make a model which is able to predict a target value by learning easy decision rules formed from the data features. There are some advantages of using this method like being easy to interpret and understand or Able to work out problems with multi-outputs. Decision Tree is a common supervised learning approach employed for both regression and classification problems. The goal of technique is forecasting a target by using easy decision rules shaped from the dataset and related features. Being easy to interpret or able to solve problems with different outputs are two advantages of using this model; on the contrary, constructing over-complex trees that cause overfitting is a typical disadvantage.



Fig(II) Decision Tree

3.1.3 Logistic Regression:

Logistic regression is used to assign observations to a separated set of classes as a classifier. The algorithm transforms its output to return a probability value with the logistic sigmoid function, and predicts the target by the concept of probability.

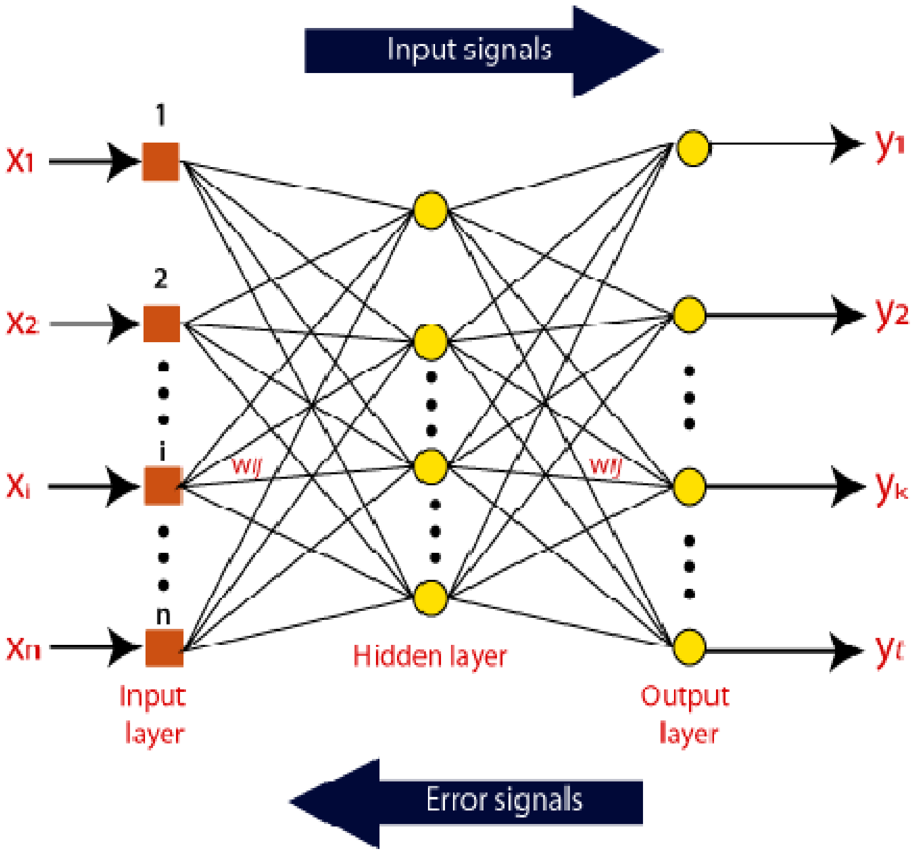
3.1.4 ANN

Artificial Neural network is typically organized in layers. Layers are being made up of many interconnected ‘nodes’ which contain an ‘activation function’. A neural network may contain the following 3 layers.

INPUT LAYER: The purpose of the input layer is to receive as input the values of the explanatory attributes for each observation. Usually, the number of input nodes in an input layer is equal to the number of explanatory variables. ‘input layer’ presents the patterns to the network, which communicates to one or more ‘hidden layers’.

HIDDEN LAYER: The Hidden layers apply given transformations to the input values inside the network. In this, incoming arcs that go from other hidden nodes or from input nodes connected to each node.

OUTPUT LAYER: The hidden layers then link to an ‘output layer‘. Output layer receives connections from hidden layers or from input layer. It returns an output value that corresponds to the prediction of the response variable.



Fig(III) Artificial Neural Network

3.2 Requirement Specifications:

HARDWARE REQUIREMENTS:

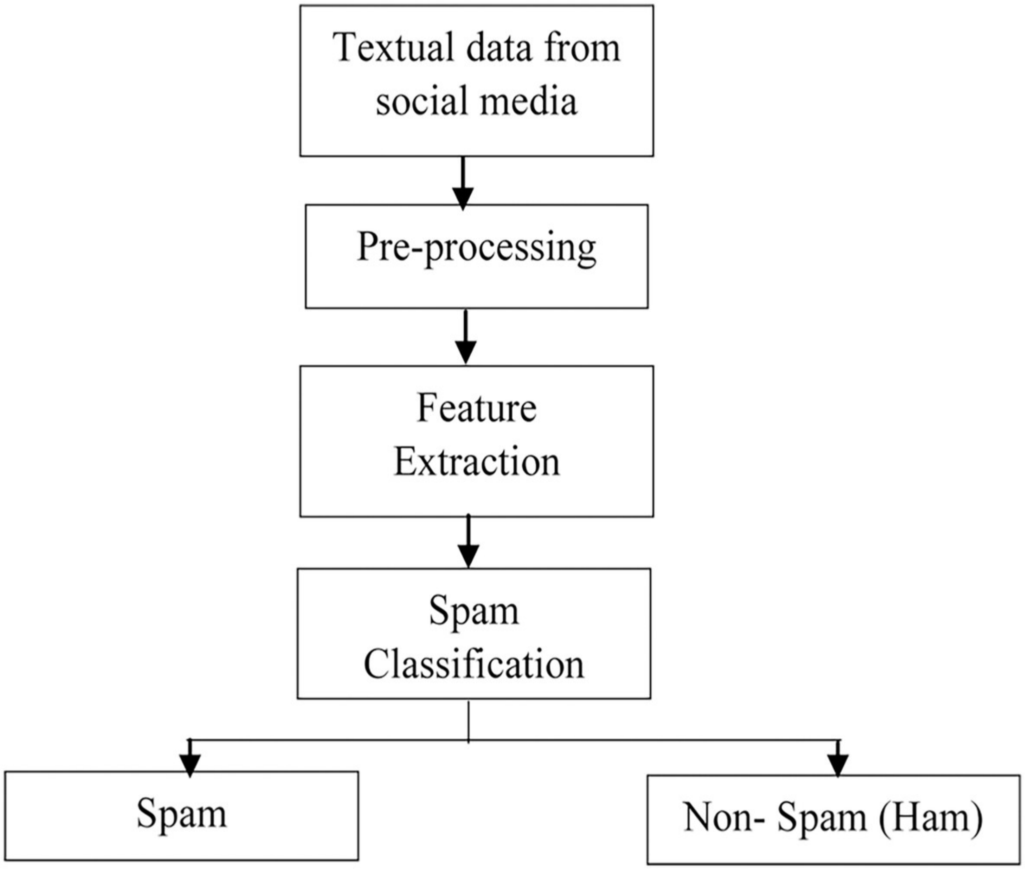
System	:	Pentium i5.
Hard Disk	:	500 GB.
Monitor	:	15’’ LED
Input Devices	:	Keyboard, Mouse
RAM	:	4 GB

SOFTWARE REQUIREMENTS:

Operating system	:	Windows 10.
Coding Language	:	Python 3.7.0

3.3 Architecture/Framework:

There are five steps in the framework detection. Some of the five steps are Data collection, Data processing, Feature Selection and feature extraction, Classification and comparison of results accuracy. These are some of the methodology for the detection framework and which helps us to provide the good result accuracy.



Fig(IV) Architecture

3.4 Algorithm and Process Design:

Data collection

Data collection is the main procedure need to be done first in this methodology detection framework. Data collection is done for the reference of machine learning which produces the prediction of future. Datasets download and collected from the UCI machine learning repository. Finally, the datasets contain 5 downloaded video from YouTube through API. The comments are taken from PSY, Katy Perry, LMFAO, Eminem, and Shakira. The total number of spam and ham comments are taken in PSY is 351, Katy Perry contains 351, LMFAO contains 439, Eminem contains 449, and Shakira contains 371.

Pre-processing

Pre-processing procedure is used for the raw datasets which will be execute the data cleaning such as tokenization, stop words removal and stemming are performed in this Pre-processing technique. Clean datasets are mainly used for the next procedure feature of selection and extraction.

Feature selection and extraction

Feature selection and extraction are the procedure which is extracted from the Pre-processing technique. This is the process before a classification class. Some of the suitable features are derived from the datasets.

Classification

Classification is a procedure which is mainly used for the training and testing process. Then out of 100, 60% is for training and another 40% is used for testing. The datasets needs are trained based on machine learning techniques.

4 Implementation and Outcome

4.1 About the data:

Data collection is the main procedure need to be done first in this methodology detection framework. Data collection is done for the reference of machine learning which produces the prediction of future. Datasets download and collected from the UCI machine learning repository. Finally, the datasets contain 5 downloaded video from YouTube through API. The comments are taken from PSY, Katy Perry, LMFAO, Eminem, and Shakira. The total number of spam and ham comments are taken in PSY is 351, Katy Perry contains 351, LMFAO contains 439, Eminem contains 449, and Shakira contains 371.

4.2 Evaluation Metrics:

F1-Score, Accuracy and Receiver Operating Characteristics-Area Under the Curve (ROC-AUC) metrics are employed to evaluate the performance of our models. For Computing F1-score and Accuracy, Precision and Recall must be evaluated by

FPR=False Positive Rate

TPR=True Positive Rate

Accuracy

Precision

Recall

F1-score

For this, the calculation of values is measured based on:

- True positive (TP) = No. of events, correctly determined.
- False negative (FN) = No. of events, inaccurately anticipated and not required.
- False-positive (FP) = No. of events, incorrectly predicted.
- True negative (TN) = No. of events, correctly anticipated and not required.

False Positive Rate (FPR): Itis a metric that can be used to assess machine learning accuracy. It is defined as:

FPR=FP/(FP+TN)

True Positive Rate (TPR): It is a synonym for recall and is therefore defined as

TPR=FP/(FP+TN)

Accuracy: It is the most important performance measure and it is easily done by a ratio of correctly predicted observations to the total observations.

Accuracy=(TN+TP)/(TP+FP+TN+FN)

Recall: It is the ratio which correctly predicts positive observations among all observations in original data.

Recall= TP/(TP+FN)

Precision: It is used to calculate the correctly identified values. This means to calculate the total number of software's which are correctly predicted as positive from the total number of software's predicted positive. It is defined as

Precision = TP/ (TP + FP)

F1-score: The F-score is a way of combining the precision and recall of the model, and it is defined as the mean of the model's precision and recall. It is also called as F-score. It is defined as

F1 Score = 2(Precision Recall/Precision + Recall)

ROC-AUC is another powerful metric for classification problems, and is calculated based on the area under ROC-AUC curve from prediction scores.

Code Implementation:

test.py

```
import pandas as pd
fromsklearn.model_selection import train_test_split
from string import punctuation
fromnltk.corpus import stopwords
importnltk
fromnltk.stem import WordNetLemmatizer
fromsklearn.feature_extraction.text import TfidfVectorizer
fromsklearn.preprocessing import LabelEncoder
fromsklearn.metrics import accuracy_score
fromsklearn import svm
importnumpy as np
fromsklearn.linear_model import LogisticRegression
fromsklearn.tree import DecisionTreeClassifier
fromsklearn.neural_network import MLPClassifier

textdata = []
labels = []
stopWords = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()
defcleanPost(doc):
tokens = doc.split()
table = str.maketrans('', '', punctuation)
tokens = [w.translate(table) for w in tokens]
tokens = [word for word in tokens if word.isalpha()]
tokens = [w for w in tokens if not w instopWords]
tokens = [word for word in tokens if len(word) > 1]
tokens = [lemmatizer.lemmatize(token) for token in tokens]
tokens = ' '.join(tokens)
return tokens

dataset = pd.read_csv('Dataset/Youtube02-KatyPerry.csv')
print(dataset.head())



for i in range(len(dataset)):
msg = dataset.get_value(i, 'CONTENT')
label = dataset.get_value(i, 'CLASS')
msg = str(msg)
msg = msg.strip().lower()
labels.append(int(label))
clean = cleanPost(msg)
textdata.append(clean)



tfidf_vectorizer = TfidfVectorizer(stop_words=stopWords, use_idf=True, smooth_idf=False,
norm=None, decode_error='replace')
tfidf = tfidf_vectorizer.fit_transform(textdata).toarray()
df = pd.DataFrame(tfidf, columns=tfidf_vectorizer.get_feature_names())
print(df.shape)
df = df.values
X = df[:, 0:df.shape[1]-1]
Y = np.asarray(labels)
indices = np.arange(X.shape[0])
np.random.shuffle(indices)
X = X[indices]
Y = Y[indices]
print(Y)
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2,random_state=0)
```

```
cls = svm.SVC(class_weight='balanced')
cls.fit(X_train,y_train)
predict = cls.predict(X_test)
acc = accuracy_score(y_test,predict)*100
print("Supervised SVM Accuracy : "+str(acc)+"\n")

lr = LogisticRegression()
lr.fit(X_train,y_train)
predict = lr.predict(X_test)
acc = accuracy_score(y_test,predict)*100
print("Logistic Regression Accuracy : "+str(acc)+"\n")

dt = DecisionTreeClassifier(max_depth = 200,class_weight='balanced')
dt.fit(X_train,y_train)
predict = dt.predict(X_test)
acc = accuracy_score(y_test,predict)*100
print("Decision Tree  Accuracy : "+str(acc)+"\n")
mlp = MLPClassifier()
mlp.fit(X_train,y_train)
predict = mlp.predict(X_test)
acc = accuracy_score(y_test,predict)*100
print("ANN  Accuracy : "+str(acc)+"\n")
```

urls.py

```
"""SpamDetection URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/2.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
```

```
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
fromdjango.contrib import admin
fromdjango.urls import path, include

urlpatterns = [
path('admin/', admin.site.urls),
path('', include('SpamDetectionApp.urls')),
]
```

Views.py

```
fromdjango.shortcuts import render
fromdjango.template import RequestContext
fromdjango.contrib import messages
fromdjango.http import HttpResponse
fromdjango.conf import settings
import pandas as pd
fromsklearn.model_selection import train_test_split
from string import punctuation
fromnltk.corpus import stopwords
importnltk
fromnltk.stem import WordNetLemmatizer
fromsklearn.feature_extraction.text import TfidfVectorizer
fromsklearn.preprocessing import LabelEncoder
fromsklearn.metrics import accuracy_score
fromsklearn import svm
importnumpy as np
fromsklearn.linear_model import LogisticRegression
fromsklearn.tree import DecisionTreeClassifier
fromsklearn.neural_network import MLPClassifier
fromsklearn.metrics import precision_score
fromsklearn.metrics import recall_score
fromsklearn.metrics import f1_score
globaltfidf_vectorizer
global classifier

stopWords = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()
```

```
defcleanPost(doc):
tokens = doc.split()
table = str.maketrans('', '', punctuation)
tokens = [w.translate(table) for w in tokens]
tokens = [word for word in tokens if word.isalpha()]
tokens = [w for w in tokens if not w instopWords]
tokens = [word for word in tokens if len(word) > 1]
tokens = [lemmatizer.lemmatize(token) for token in tokens]
tokens = ' '.join(tokens)
return tokens


def index(request):
ifrequest.method == 'GET':
return render(request, 'index.html', {})

defBuildSpamDetector(request):
ifrequest.method == 'GET':
return render(request, 'BuildSpamDetector.html', {})

Def SpamDetection(request):
ifrequest.method == 'GET':
return render(request, 'SpamDetection.html', {})


Def BuildSpamDetectorAction(request):


ifrequest.method == 'POST':
globaltfidf_vectorizer
global classifier
name = request.POST.get('t1', False)
dataset = pd.read_csv('Dataset/'+name)
print(dataset.head())
textdata = []
labels = []
for i in range(len(dataset)):
msg = dataset._get_value(i, 'CONTENT')
label = dataset._get_value(i, 'CLASS')
msg = str(msg)
msg = msg.strip().lower()
labels.append(int(label))
clean = cleanPost(msg)
textdata.append(clean)
tfidf_vectorizer = TfidfVectorizer(stop_words=stopWords, use_idf=True, smooth_idf=False,
norm=None, decode_error='replace')
tfidf = tfidf_vectorizer.fit_transform(textdata).toarray()
```

```
df = pd.DataFrame(tfidf, columns=tfidf_vectorizer.get_feature_names())
print(df.shape)
df = df.values
        X = df[:, 0:df.shape[1]-1]
        Y = np.asarray(labels)
indices = np.arange(X.shape[0])
np.random.shuffle(indices)
        X = X[indices]
        Y = Y[indices]
print(Y)
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2,random_state=0)
cls = svm.SVC(class_weight='balanced')
cls.fit(X,Y)
prediction_data = cls.predict(X_test)
acc = accuracy_score(y_test,prediction_data)*100
 p = precision_score(y_test, prediction_data,average='macro') * 100
        r = recall_score(y_test, prediction_data,average='macro') * 100
        f = f1_score(y_test, prediction_data,average='macro') * 100
color = '<font size="" color="black">'
output = '<table border="1" align="center"><tr><th>Algorithm
Name</th><th>Accuracy</th><th>Precision</th><th>Recall</th><th>F-SCORE</th></tr>'
        output+='<tr><td>'+color+'SVM
Algorithm</td><td>'+color+str(acc)+'</td><td>'+color+str(p)+'</td><td>'+color+str(r)+'</td><t
d>'+color+str(f)+'</td></tr>'
```

```
lr = LogisticRegression()
lr.fit(X_train,y_train)
prediction_data = lr.predict(X_test)
acc = accuracy_score(y_test,prediction_data)*100
        p = precision_score(y_test, prediction_data,average='macro') * 100
        r = recall_score(y_test, prediction_data,average='macro') * 100
        f = f1_score(y_test, prediction_data,average='macro') * 100
        output+='<tr><td>'+color+'Logistic Regression
Algorithm</td><td>'+color+str(acc)+'</td><td>'+color+str(p)+'</td><td>'+color+str(r)+'</td><t
d>'+color+str(f)+'</td></tr>'


dt = DecisionTreeClassifier(max_depth = 200,class_weight='balanced')
dt.fit(X_train,y_train)
prediction_data = dt.predict(X_test)
acc = accuracy_score(y_test,prediction_data)*100
        p = precision_score(y_test, prediction_data,average='macro') * 100
        r = recall_score(y_test, prediction_data,average='macro') * 100
        f = f1_score(y_test, prediction_data,average='macro') * 100
        output+='<tr><td>'+color+'Decision Tree
Algorithm</td><td>'+color+str(acc)+'</td><td>'+color+str(p)+'</td><td>'+color+str(r)+'</td><t
d>'+color+str(f)+'</td></tr>'
classifier = dt
mlp = MLPClassifier()
mlp.fit(X_train,y_train)
prediction_data = mlp.predict(X_test)
acc = accuracy_score(y_test,prediction_data)*100
        p = precision_score(y_test, prediction_data,average='macro') * 100
        r = recall_score(y_test, prediction_data,average='macro') * 100
        f = f1_score(y_test, prediction_data,average='macro') * 100
        output+='<tr><td>'+color+'ANN
Algorithm</td><td>'+color+str(acc)+'</td><td>'+color+str(p)+'</td><td>'+color+str(r)+'</td><t
d>'+color+str(f)+'</td></tr>'

context= {'data':output}
return render(request, 'ViewResult.html', context)

defSpamDetectionAction(request):
ifrequest.method == 'POST':
globaltfidf_vectorizer
global classifier
comment = request.POST.get('t1', False)
review = comment.lower()
review = review.strip().lower()
review = cleanPost(review)
```

```
testReview = tfidf_vectorizer.transform([review]).toarray()
testReview = testReview[:, 0:testReview.shape[1]-1]
predict = classifier.predict(testReview)
print(predict)
msg = 'none'
if predict == 0:
msg = 'The given comments are NOT A SPAM. (It is a HAM)'
if predict == 1:
msg = 'The given comments are NOT A HAM. (It is a SPAM)'
context= {'msg':msg}
return render(request, 'SpamDetection.html', context)
```

Index.html

```
{% load static %}
<html>
<head>
<title>Spam Detection for Youtube Comments</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="{% static 'style.css' %}" rel="stylesheet" type="text/css" media="screen" />
</head>
<body>
<div id="wrapper">
    <div id="header">
        <div id="logo">
            <center><font size="4" color="yellow">Spam Detection for Youtube
Comments</font></center>
        </div>
        <div id="slogan">

        </div>
    </div>
    <div id="menu">
        <ul><center>
    <li><a href="{% url 'index' %}">Home</a></li>
    <li><a href="{% url 'BuildSpamDetector' %}">Build Spam Detection Machine Learning
Models</a></li>
        <li><a href="{% url 'SpamDetection' %}">Detect Spam from Comments</a></li>
        </center></ul>
<br class="clearfix" />
                </div>

    <div id="splash">
        <img class="pic" src="static/images/investor.jpg" width="870" height="230" alt="" />
    </div>
            <center>
    <br/>
```

```
<h3><b><font size="" color="black">Spam Detection for Youtube Comments</font></b></h3>
   {% csrf_token %}

                              <br/><br/><br/><br/><br/><br/>

        </body>
</html>
```

Style.CSS

```
/*
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 3.0 License

Name       : Big Business
Description: A two-column, fixed-width design with a bright color scheme.
Version    : 1.0
Released   : 20120210
*/

* {
        margin: 0;
        padding: 0;
}

a {
        text-decoration: underline;
        color: #0F8C8C;
}

a:hover {
        text-decoration: none;
}

body {
        font-size: 11.5pt;
        color: #5C5B5B;
        line-height: 1.75em;
        background: #E0DCDC url(images/img01.gif) repeat-x top left;
}

body,input {
        font-family: Georgia, serif;
}
```

```
strong {
	color: #2C2B2B;
}

br.clearfix {
	clear: both;
}

h1,h2,h3,h4 {
	font-weight: normal;
	letter-spacing: -1px;
}
h2 {
	font-size: 2.25em;
}

h2,h3,h4 {
	color: #2C2B2B;
	margin-bottom: 1em;
}

h3 {
	font-size: 1.75em;
}

h4 {
	font-size: 1.5em;
}

img.alignleft {
	margin: 5px 20px 20px 0;
	float: left;
}

img.aligntop {
	margin: 5px 0 20px 0;
}
img.pic {
	padding: 5px;
	border: solid 1px #D4D4D4;
}

p {
	margin-bottom: 1.5em;
}
```

```
ul {
        margin-bottom: 1.5em;
}

ul h4 {
        margin-bottom: 0.35em;
}

.box {
        overflow: hidden;
        margin-bottom: 1em;
}

.date {
        background: #6E6E6E;
        padding: 5px 6px 5px 6px;
        margin: 0 6px 0 0;
color: #FFFFFF;
        font-size: 0.8em;
        border-radius: 2px;
}

#content {
        width: 665px;
        float: left;
        padding: 0;
}

#content-box1 {
        width: 320px;
        float: left;
}

#content-box2 {
        margin: 0 0 0 345px;
        width: 320px;
}

#footer {
        margin: 40px 0 120px 0;
        text-align: center;
        color: #8C8B8B;
}

#footer a {
```

```
color: #8C8B8B;
}

#header {
	height: 75px;
	position: relative;
	background: #6E6E6E url(images/img03.jpg) top left no-repeat;
	padding: 45px;
	color: #FFFFFF;
	width: 888px;
border: solid 1px #7E7E7E;
	border-top-left-radius: 5px;
	border-top-right-radius: 5px;
	overflow: hidden;
}

#logo {
	line-height: 160px;
	height: 160px;
	padding: 5px 0 0 0;
	position: absolute;
	top: 0;
	left: 45px;
}

#logo a {
	text-decoration: none;
	color: #FFFFFF;
	text-shadow: 0 1px 1px #3E3E3E;
}

#logo h1 {
	font-size: 2.25em;
}
#slogan {
	line-height: 160px;
	height: 160px;
	padding: 5px 0 0 0;
	position: absolute;
	right: 45px;
	top: 0;
}

#slogan h2 {
	color: #BEBEBE;
	font-size: 1.25em;
```

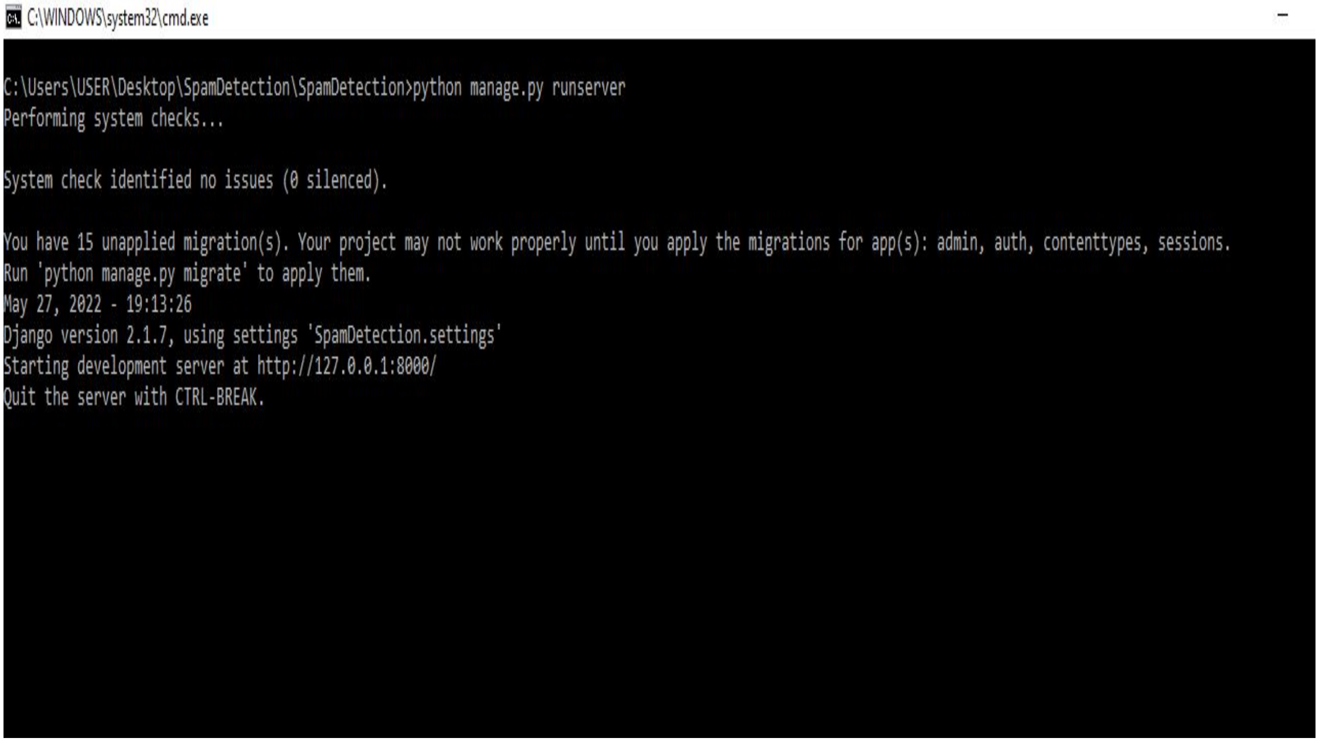
```
        text-shadow: 0 1px 1px #3E3E3E;
}

#menu {
        padding: 0 45px 0 45px;
        position: relative;
        background: url(images/img01.gif) repeat-x top left;
        margin: 0 0 0 0;
        height: 60px;
        line-height: 60px;
        width: 890px;
        border-top: solid 1px #5AD7D7;
        text-shadow: 0 1px 1px #007D7D;
}

#menu a {
        text-decoration: none;
        color: #FFFFFF;
        font-size: 1.25em;
        letter-spacing: -1px;
}

#menu ul {
        list-style: none;
}

#menu ul li {
        padding: 0 20px 0 20px;
        display: inline;
}

#menu ulli.first {
        padding-left: 0;
}

#page {
        padding: 45px 45px 15px 45px;
        position: relative;
        width: 890px;
        margin: 0;
}

#page .section-list {
        list-style: none;
        padding-left: 0;
}
```

```
#page .section-list li {
        clear: both;
        padding: 30px 0 30px 0;
}
0
#page ul {
        list-style: none;
}

#page ul li {
        border-top: solid 1px #D4D4D4;
        padding: 15px 0 15px 0;
}

#page ulli.first {
        padding-top: 0;
        border-top: 0;
}

#page-bottom {
        position: relative;
        margin: 0;
        background: #6E6E6E url(images/img03.jpg) top left no-repeat;
        border: solid 1px #7E7E7E;
        width: 888px;
        padding: 45px 45px 0 45px;
        color: #DCDCDC;
        border-bottom-left-radius: 5px;
        border-bottom-right-radius: 5px;
}

#page-bottom a {
        color: #F5F5F5;
}

#page-bottom h2, #page-bottom h3, #page-bottom h4 {
        color: #FFFFFF;
}
#page-bottom ul {
        list-style: none;
}

#page-bottom ul li {
        border-top: solid 1px #8F8F8F;
        padding: 15px 0 15px 0;
}
```

```
#page-bottom ulli.first {
	padding-top: 0;
	border-top: 0;
}

#page-bottom-content {
	width: 665px;
	float: left;
}

#page-bottom-sidebar {
	width: 200px;
	margin: 0 0 0 690px;
}

#search input.form-submit {
	margin-left: 1em;
	color: #FFFFFF;
	padding: 10px;
	background: #2FACAC;
	border: 0;
}

#search input.form-text {
	border: solid 1px #8F8F8F;
	padding: 10px;
}

#sidebar {
width: 200px;
	padding: 0;
	margin: 0 0 0 690px;
}

#splash {
	margin: 0 0 0 0;
	height: 250px;
	position: relative;
	padding: 45px 45px 10px 45px;
	width: 890px;
}

#splash .pic {
	padding: 9px;
}
```

```
#wrapper {
	position: relative;
	width: 980px;
	margin: 75px auto 0 auto;
	background: #FFFFFF;
}
```

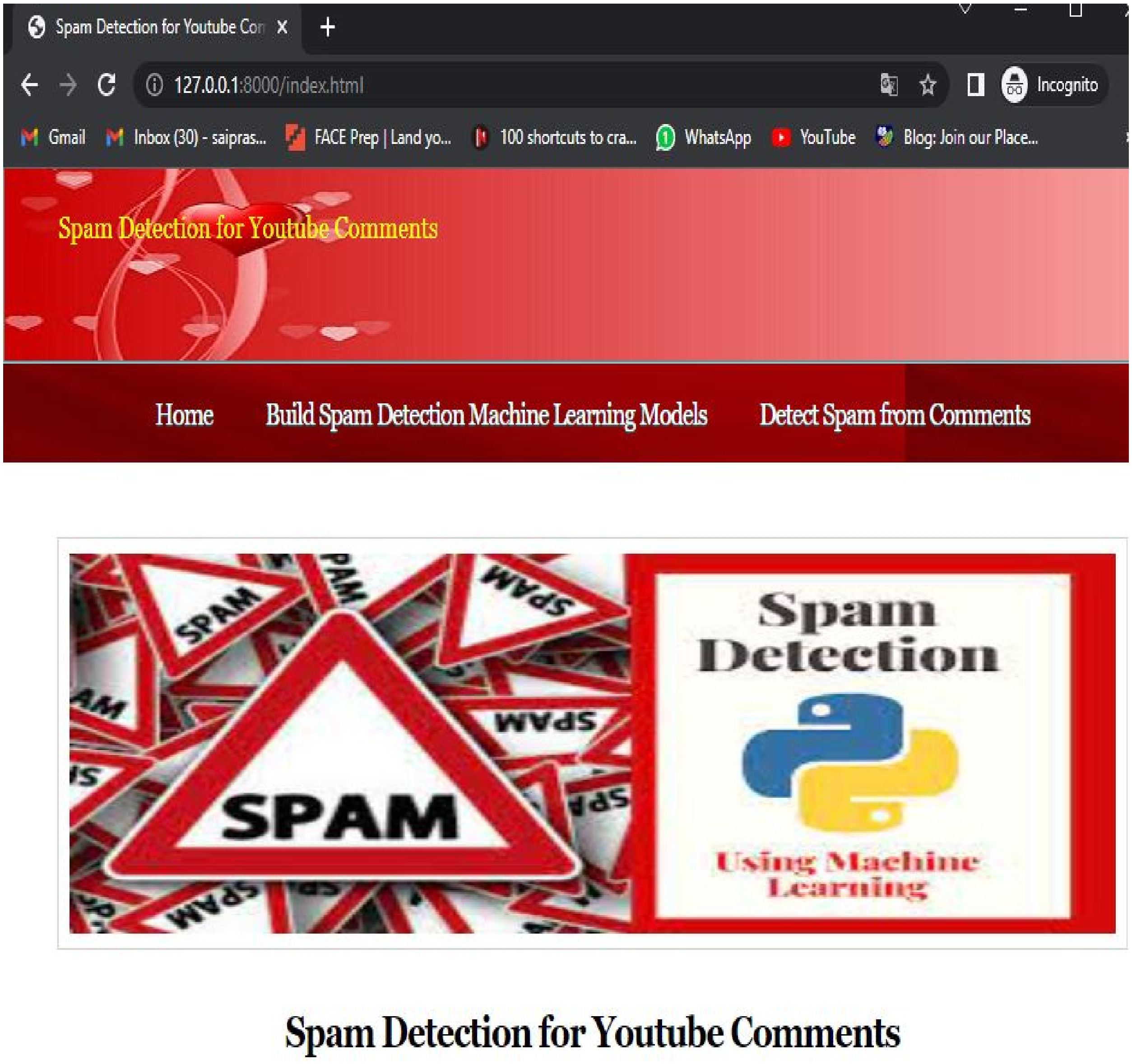
SCREENSHOT :



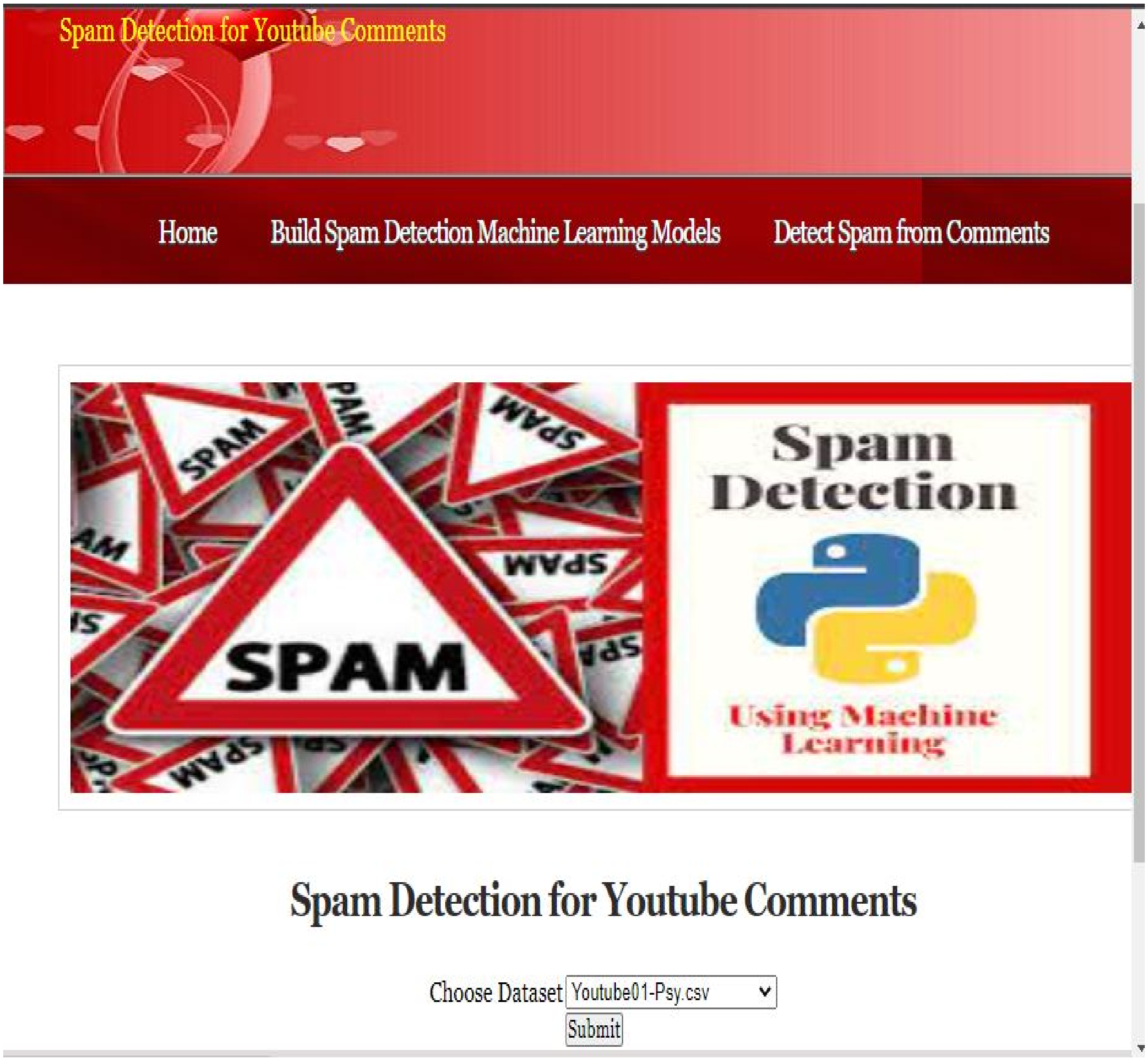
C:\WINDOWS\system32\cmd.exe

```
C:\Users\USER\Desktop\SpamDetection\SpamDetection>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
May 27, 2022 - 19:13:26
Django version 2.1.7, using settings 'SpamDetection.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Fig(V): spam detection index.html


Spam Detection for Youtube Con
127.0.0.1:8000/index.html
Incognito
Gmail
Inbox (30) - saipras...
FACE Prep | Land yo...
100 shortcuts to cra...
WhatsApp
YouTube
Blog: Join our Place...
Spam Detection for Youtube Comments
Home
Build Spam Detection Machine Learning Models
Detect Spam from Comments
SPAM
Spam Detection
Using Machine Learning
Spam Detection for Youtube Comments

Fig(VI) Spam Detection For Youtube Comments

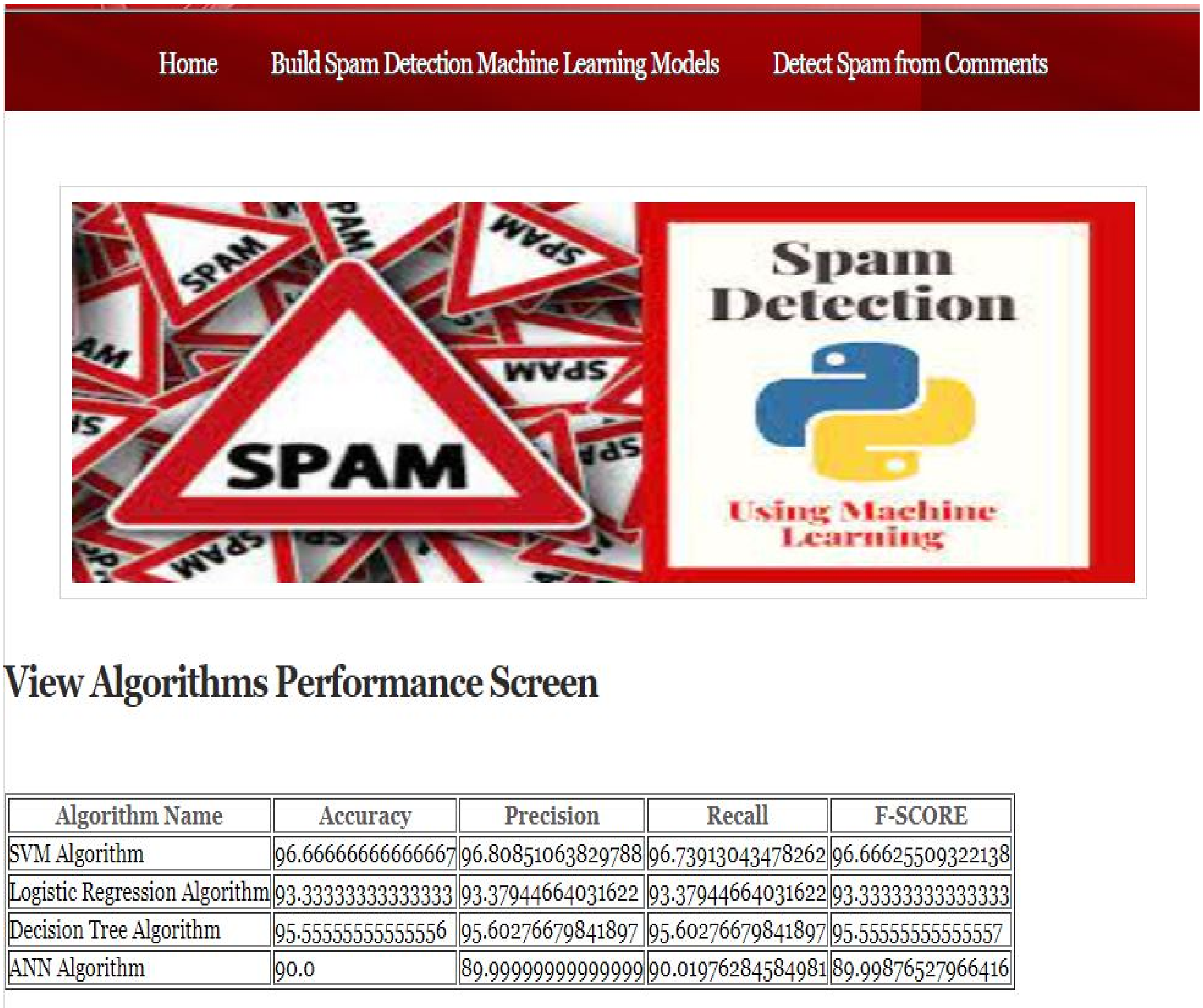


Spam Detection for Youtube Comments
Home
Build Spam Detection Machine Learning Models
Detect Spam from Comments
SPAM
Spam Detection
Using Machine Learning

Spam Detection for Youtube Comments

Choose Dataset Youtube01-Psy.csv

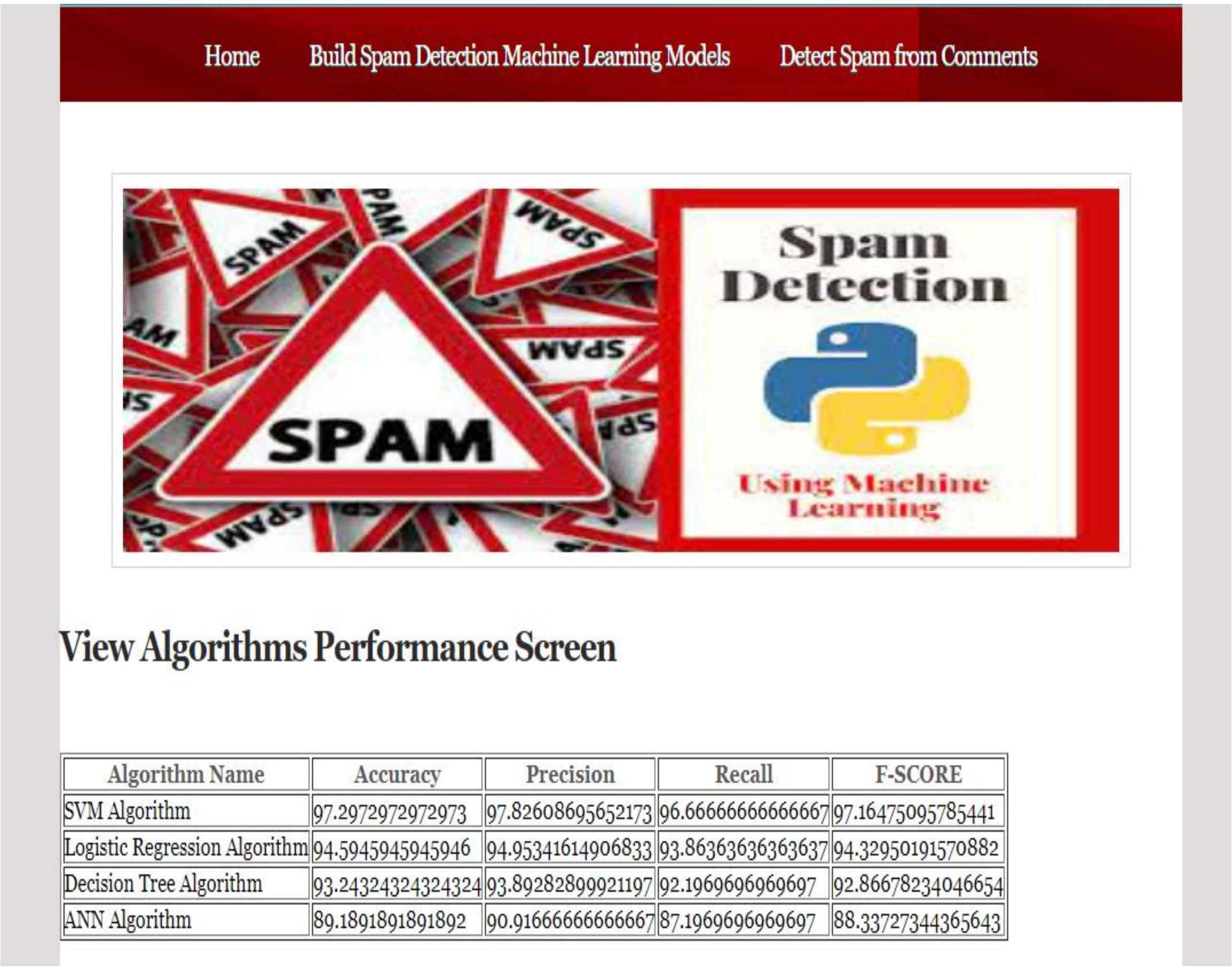
Fig(VII) Choose Dataset


Home
Build Spam Detection Machine Learning Models
Detect Spam from Comments
SPAM
Spam Detection
Using Machine Learning

View Algorithms Performance Screen

Algorithm Name	Accuracy	Precision	Recall	F-SCORE
SVM Algorithm	96.66666666666667	96.80851063829788	96.73913043478262	96.66625509322138
Logistic Regression Algorithm	93.33333333333333	93.37944664031622	93.37944664031622	93.33333333333333
Decision Tree Algorithm	95.55555555555556	95.60276679841897	95.60276679841897	95.55555555555557
ANN Algorithm	90.0	89.99999999999999	90.01976284584981	89.99876527966416

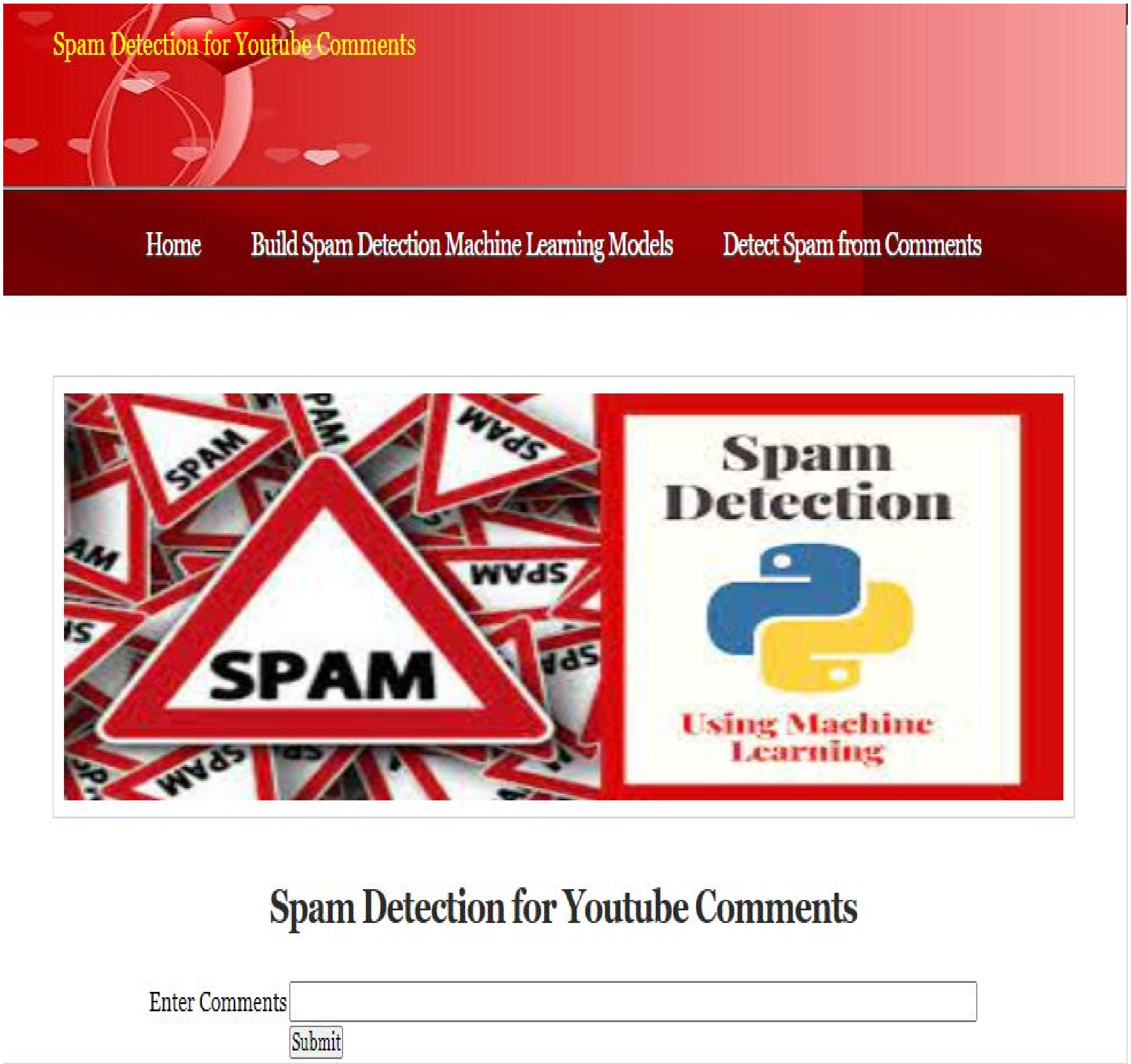
Table(I) Algorithm Performance


Home
Build Spam Detection Machine Learning Models
Detect Spam from Comments
SPAM
Spam Detection
Using Machine Learning

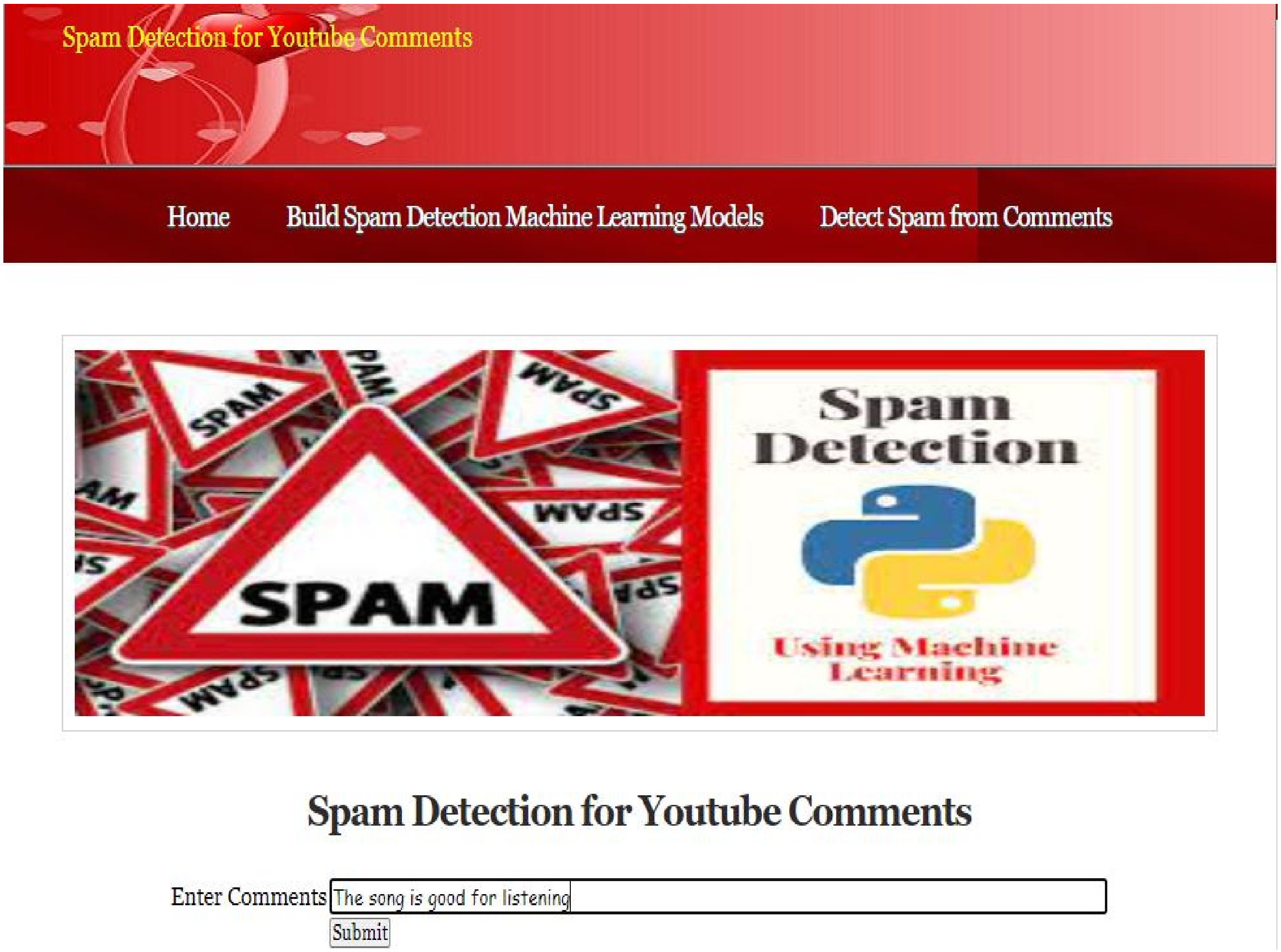
View Algorithms Performance Screen

Algorithm Name	Accuracy	Precision	Recall	F-SCORE
SVM Algorithm	97.2972972972973	97.82608695652173	96.66666666666667	97.16475095785441
Logistic Regression Algorithm	94.5945945945946	94.95341614906833	93.86363636363637	94.32950191570882
Decision Tree Algorithm	93.24324324324324	93.89282899921197	92.1969696969697	92.86678234046654
ANN Algorithm	89.1891891891892	90.91666666666667	87.1969696969697	88.33727344365643

Table(II) Algorithm Performance Screen


Spam Detection for Youtube Comments
Home
Build Spam Detection Machine Learning Models
Detect Spam from Comments
SPAM
Spam Detection
Using Machine Learning
Spam Detection for Youtube Comments
Enter Comments
Submit

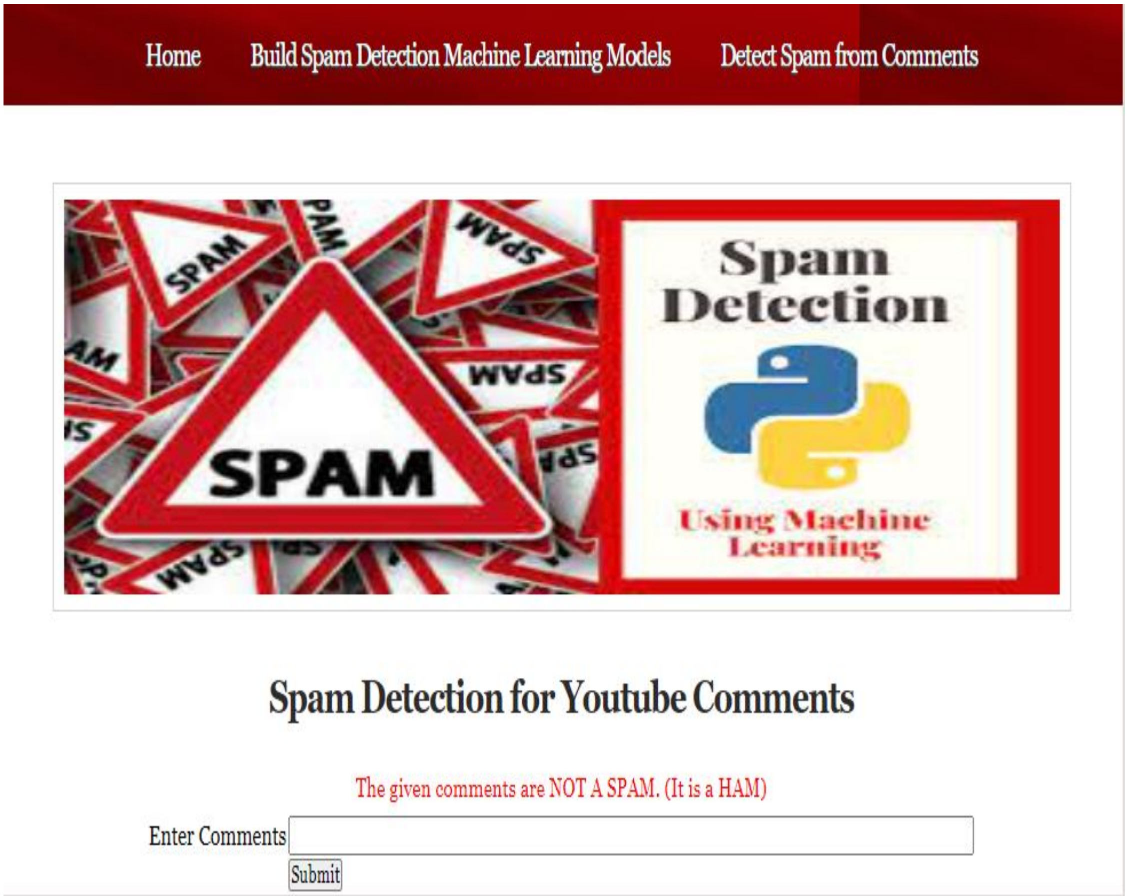
Fig(VIII): Spam Detection for youtube comments


Spam Detection for Youtube Comments
Home
Build Spam Detection Machine Learning Models
Detect Spam from Comments
SPAM
Spam Detection
Using Machine Learning
Submit

Spam Detection for Youtube Comments

Enter Comments: The song is good for listening

Fig(IX): Youtube Spam Detection Comments


Home
Build Spam Detection Machine Learning Models
Detect Spam from Comments
SPAM
Spam Detection
Using Machine Learning
Enter Comments
Submit

Spam Detection for Youtube Comments

The given comments are NOT A SPAM. (It is a HAM)

Fig(X) Given Comment Spam or Ham

4.3 Outcome:

In this project author is using UCI machine learning youtube dataset from 5 different videos to build spam detection machine learning models. In this project author has used only SM algorithm and we have added decision tree and logistic regression and ANN algorithm and then calculating accuracy, precision, recall and FSCORE between all algorithms and this application will split dataset into train and test where 80% dataset will be used for training and 20% will be used for testing and this dataset split is based on random so accuracy may vary during each execution.

To run project install python 3.7 and DJANGO and SKLEARN package and then double click on 'run.bat' file to start DJANGO server and then run code from browser by entering URL as 'http://127.0.0.1:8000/index.html' and press enter key to get home page

In above screen DJANGO server started and now run in browser

In above screen click on 'Build Spam Detection Machine Learning Models' link to get below screen to generate machine learning models

In above screen select desired dataset and then click on 'Submit' button to train selected dataset with SVM, logistic regression, decision tree and ANN algorithm and then calculate accuracy

In above screen I selected first dataset and then click on 'Submit' button to get below output

In above screen we trained on all algorithms on selected dataset and then calculate above metrics and from above values SVM is giving better result and now ML models are ready and we can go for prediction by clicking on 'Detect Spam from Comments' link to get below screen
In above screen enter you comments and then click on 'Submit' button to get below result

In above screen I entered some comments and below is the result

In above screen in red colour message we got prediction result as ‘given comments is HAM not SPAM’ and now test other message

For above message below is the result

Above comments predicted as SPAM. Similarly you can give any sentence and get prediction result.
Note: ML will predict only those messages as SPAM or HAM which are given in dataset as SPAM or HAM

CONCLUSION

In this project we using UCI Machine Learning YouTube datasets from 5 different videos to build spam detection machine learning models. In this project we used only SVM algorithm, we have added decision tree and logistic regression and ANN algorithm and then calculating accuracy, precision, recall and FSCORE. Between all algorithms and this application will split dataset into train and test where 80% dataset will be used for training and 20% will be used for testing and this dataset split is based on random so accuracy may vary during each execution. We trained on all algorithms on selected dataset and then calculate above metrics and from above values SVM is giving better result. Machine Learning will predict only those messages as SPAM or HAM which are given in dataset.

Bibliography

1. Aqliima Aziz1, CikFeresaMohd Foozy2, Palaniappan Shamala3, Zurinah Suradi4 YouTube spam comment detection using support vector machine and K–nearest neighbour. (ACT) Faculty of Computer Science and Information Technology, (UTHM).

2 Dr.E.Srinivasa Reddy a Survey on Spam Detection Methodologies in Social Networking Sites, Research Scholar, Computer Science and EngineeringAcharyaNagarjuna University, Guntur, AP, India.

3. Tube Spam: Comment Spam Filtering on YouTube ,T´ulio C. Alberto, Johannes V. Lochter, Tiago A. AlmeidaDepartment of Computer ScienceFederal University.

4. How Useful Are Your Comments? Analyzing and Predicting YouTube Comments and Comment Ratings. Stefan Siersdorfer, SergiuChelaru,WolfgangNejdl L3S Research Center.

5. Depth Analysis of User Comments on YouTube.

6. Manwar, S. R., Lambhate, P., &Patil, J. (2017). Classification Methods for Spam Detection Online Social Network.