**CHAPTER 1**

# 1.INTRODUCTION

In this project,customers can post their reviews or opinions on several websites.These reviews are helpful for the organizations and for future customers,who get an idea about products or services before making a selection.In recent years,it has observed that the number of customer reviews has increased significantly,Customers reviews affect the decision of potential buyers.In other words,when customers see reviews on social media,they determine whether customers to buy the product or reverse their purchasing decisions.Therefore.consumer reviews offer an invaluable service for individuals.

The way consumers openly express and use their feedback has contributed to issues with websites containing customer reviews. Social media (Twitter, Facebook, etc.) allows anyone to freely post feedback or criticism of any company at any time with no obligations or limits. The lack of restrictions, in turn, leads certain companies to use social media to unfairly promote their goods, brands or shops, or to unfairly criticize those of their rivals. For example, suppose a few consumers who bought a specific digital camera posted negative reviews on image quality. These reviews portray the digital camera unfavorably to the public. Thus, the camera manufacturer might employ an individual or team to post fake positive reviews about the camera. Similarly, in order to promote the company, the producer might ask the hired persons to post negative comments about competitors' products. Reviews published by people who have not personally encountered the items being reviewed are considered fake reviews . Accordingly, a person who posts fake reviews is called a spammer.When the spammer works with other spammers to achieve a specific goal, the spammers are called a group of spammers. Many studies have investigated the fake review detection problem and its challenges. The main task associated with fake review detection is classifying the review as fake or genuine. In this survey paper, we have presented a comprehensive survey of the literature to further identify existing problems for future directions in this research area. It provides traditional statistical machine learning and deep learning techniques which will assist researchers, who are interested in fake review detection, to choose the best machine learning method. To help the reader easily understand the field of fake review detection, relevant publications from Google Scholar,Web of Sciences, and some high-profile conferences are presented in this paper to demonstrate the challenges in the field.

## 1.1 EXISTING SYSTEM

Content based methods focus on what is the content of the review, and have attempted to detect spam review by analyzing the linguistic features they used three techniques to perform

classification. These three techniques are- genre identification, detection of psycholinguistic deception,text categorization.Behavior feature based study focuses on the reviewer that includes characteristics of the person who is giving the review. addressed the problem of review spammer detection, or finding users who are the source of spam reviews. People who post intentional fake reviews have significantly different behavior than the normal user. They have identified the following deceptive rating and review behaviors.Deceptive online review detection is generally considered as a classification problem and one popular approach is to use supervised text classification techniques. These techniques are robust if the training is performed using large datasets of labeled instances from both classes, deceptive opinions (positive instances) and truthful opinions (negative examples). Some researchers also used semi-supervised classification techniques.

**1.2 PROPOSED SYSTEM:**

In the proposed system,each review goes through tokenization process first.Then,unnecessary words are removed and candidate feature words are generated.Each candidate feature words are checked against the dictionary and if its entry is available in the dictionary then its frequency is counted and added to the column in the feature vector that corresponds the numeric map of the word.Alongside with counting frequency, the length of the review is measured and added to the feature vector.Finally, sentiment score which is available in the data set is added in the feature vector. We have assigned negative sentiment as zero valued and positive sentiment as some positive value in the feature vector. The system proposed will include methods like collection of datasets from Kaggle and preprocessing them.

**Pre-Processing**

The term Pre-processing the data is defined as the process of converting a data into an understandable format by cleaning it and preparing the text for classification. Texts from online contain usually lots of noise and uninformative parts such as scripts and advertisements. Processing includes certain steps such as online text cleaning, white space removal, expanding

abbreviation, Stemming, stop words removal and feature selection. These might reduce the noise in the text which helps to speed up the performance of the classifier. Before carrying out

the transformation and vectorization of the sentences of the reviews, preprocessing steps were used to clean the data and remove noise. The goal of text pre- processing is to convert the texts of the reviews to a form that deep learning algorithms can understand and analyze. The pre-processing steps are as follows: a) Removing punctuation: deleting punctuation marks from the reviews. b) Removing stop words: This process cleans articles from the text; for example,

‘the’, ‘a’, ‘’ words are removed from text. c) Stripping useless words and characters from the dataset. d) Word stemming: converting each word of a sentence into its root; for instance, ‘undesired’ becomes ‘desire’ e) Tokenizing: splitting whole sentences in the text into separate words, keywords, phrases, and pieces of tokens. f) Padding sequences: using deep learning neural networks to ensure that the inputdata have equal sequence length. However, we implemented a pre-padding method to add zeros to the beginnings of the vector representation.

Understanding deviation of ratings: -

* The ratings or reviews which are showing a trend of continuous growth but suddenly shows negativity is simply displaying a deviation from the normal ratings.

Sentiment analysis of the product review: -

It is necessary for the system to understand whether the review is positive or negative, which further helps to understand the deviation from either the positivity or the negativity in the reviews. The analysis will help us to understand the overall aspect of the products so that few spam reviews doesn’t affect the overall statistics of products.

* The posted reviews will undergothe process of sentiment analysis, IP address track, and its deviation from overall reviews. Incase of miscalculations, reviews will be analyzed and detected.

Web Scripting is an automatic method to obtain large amounts of data fromwebsites. Most of this data is unstructured data in an HTML format which is thenconverted into structured data in a spreadsheet or a database so that it can be used in various applications. This large amounts of data from a website are used to train an algorithm. Web scraping requires two parts namely the crawler and the scraper. The crawler is an artificial intelligence algorithm that browses the web to search the data required by following the links across the internet. The scraper, on the other hand, is a specific tool created to extract the data from the website. The design of the

scraper can vary greatly according to the complexity and scope of the project so that it can quickly and accurately extract the data. When a web scraper needs to scrape a site, first it is provided the URLs of the required sites. Then it loads all the HTML code from those sites and a more advanced scraper might even extract all the CSS and JavaScript elements as well. Then the scraper obtains the required data from this HTML code and outputs this data in the format specified by the user. Initially, a website is created which contains featured products of famous brands. Users have to login to the website for entering reviews. Once the reviews have been entered, machine learning algorithms will be used for classifying them into fake or real. Fake or spam reviews will be removed thereafter from the website. Only thereviews which remain

truthful gets published in this process. Thus, the product review website is an efficient and effective way for users to know about the actual information of the product.

B. We Are Using Two Machine Learning Algorithms

1. TF-IDF Vectorizer: TF-IDF Vectorizer (Term FrequencyInverter Document Frequency): TF-IDF which stands for Term Frequency– Inverse Document Frequency is a statistical method of evaluating the significance of word in given documents. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction. TF- IDF vectorizer is defined with parameter (stop words= ‘English’) which eliminates all the common English words.
2. *Naïve Bayes Classifier:* Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts based on the probability of an object. It is called Bayes because it depends on the principle of Bayes theorem, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability. Naïve Bayes Classifier works on the following steps:

* Convert the given dataset into frequency tables. Generate Likelihood table by finding the probabilities of given features. Now, use Bayes theorem to calculate the posterior probability. Formula: P (c|x) = P(x|c) P(c) / P(x) Referred from Bayes's theorem, in probability theory, a means for revising predictions considering relevant evidence, also known as conditional probability or inverse probability.
* Passive Aggressive Classifier Passive-Aggressive algorithms are called so because Passive- If the prediction is correct, keep the representation and do not make any interchanges. i.e., the data in the example is not enough to cause any changes in the representation. Aggressive- If the prediction is incorrect, make interchanges to the representation.

i.e., some interchange to the representation may correct it. Understanding the mathematics supporting this algorithm is not very simple and is supporting the scope of a single article. This section provides just an overview of the algorithm and a simple implementation of it. To learn more about the mathematics supporting this algorithm.

**CHAPTER 2**

# 2. LITERATURE SURVEY

**A.Revisiting Semi-Supervised Learning for Online Deceptive Review Detection:**

With more consumers using online opinion reviews to inform their service decision making, opinion reviews have an economical impact on the bottom line of businesses. Unsurprisingly, opportunistic individuals or groups have attempted to abuse or manipulate online opinion reviews (e.g., spam reviews) to make profits and so on, and that detecting deceptive and fake opinion reviews is a topic of ongoing research interest. In this paper, we explain how semi-supervised learning methods can be used to detect spam reviews, prior to demonstrating its utility using a data set of amazon reviews.

**B. Detecting product review spammers using rating behaviors:**

This paper aims to detect users generating spam reviews or review spammers. We identify several characteristic behaviors of review spammers and model these behaviors so as to detect the spammers. In particular, we seek to model the following behaviors. First, spammers may target specific products or product groups in order to maximize their impact. Second, they tend to deviate from the other reviewers in their ratings of products. We propose scoring methods to measure the degree of spam for each reviewer and apply them on an Amazon review dataset. We then select a subset of highly suspicious reviewers for further scrutiny by our user evaluators with the help of a web based spammer evaluation software specially developed for user evaluation experiments. Our results show that our proposed ranking and supervised methods are effective in discovering spammers and outperform other baseline methods based on helpfulness votes alone. We finally show that the detected spammers have a more significant impact on ratings compared with the unhelpful reviewers.

**C.Towards a general rule for identifying deceptive opinion spam:**

Consumers purchase decisions are increasingly influenced by user-generated online reviews. Accordingly, there has been growing concern about the potential for posting deceptive opinion spam - fictitious reviews that have been deliberately written to sound authentic, to deceive the reader. In this paper, we explore generalized approaches for identifying online deceptive opinion spam based on a new gold standard dataset, which is

comprised of data from three different domains (i.e. Hotel, Restaurant, Doctor), each of which contains three types of reviews.

**D.****Detection of review spam:**

In recent years, online reviews have become the most important resource of customers’ opinions. These reviews are used increasingly by individuals and organizations to make purchase and business decisions. Unfortunately, driven by the desire for profit or publicity, fraudsters have produced deceptive (spam) reviews.

The fraudsters’ activities mislead potential customers and organizations reshaping their businesses and prevent opinion-mining techniques from reaching accurate conclusions. The present research focuses on systematically analyzing and categorizing models that detect review spam. Next, the study proceeds to assess them in terms of accuracy and results. We find that studies can be categorized into three groups that focus on methods to detect spam reviews, individual spammers and group spam. Different detection techniques have different strengths and weaknesses and thus favor different [detection contexts.](https://www.sciencedirect.com/topics/computer-science/context-detection)

# CHAPTER 3

## 3.REQUIREMENT SPECIFICATIONS

### 3.1 SOFTWARE REQUIREMENTS

Technology : Python

Web Server : FLASK

Editor : Pycharm

### 3.2 HARDWARE REQUIREMENTS

System : i3

Hard Disk : 40 GB

Floppy Drive :1.44 Mb

### 3.3 FUNCTIONAL REQUIREMENTS

●  **User Interfaces:**

In this module, users need to upload an image containing text in appropriate format. Appropriate

error handling is done using exceptions in-order to isolate abnormal results or conditions. A well-connected internet connection either using a modem or cable or Wi-Fi or any other form should exist.

The client only requires a browser for communication.

## ● Software Interface

The input to the system would be data in the form of an image containing text it. The outgoing data would be the sentiment polarities of the text extracted fromthe image uploaded. A compatible browser is required to access the data in the form of an image from the client.

### 3.4 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to

judge the operation of a system rather than specific behaviour. They may relate to emergent

system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:

* Product Requirements
* Organizational Requirements
* Reliability
* Performance Requirements

# CHAPTER 4

## 4.IMPLEMENTATION

In this system overview we will undergo various modules for implementing our project they are listed below:

**1.Data Collection:**

Using this module we will upload AMAZON reviews dataset to the application.

**2.Data Preprocessing:**

Using this module we will read all reviews and then remove stop words, special symbols, punctuation and numeric data from all reviews and after applying Preprocessing we will extract features from all reviews.

**3.Features Extraction:**

Here we will apply the TF-IDF (term frequency Inverse Document Frequency) algorithm to convert string reviews into numeric vectors. Each word count will be put in a vector in place of words.

**4.Run SVM Algorithm:**

We will apply SVM algorithm on TF-IDF vector to train SVM algorithm and then we apply test data on SVM trained model to calculate SVM prediction accuracy

**5.Run Naive Bayes Algorithm:**

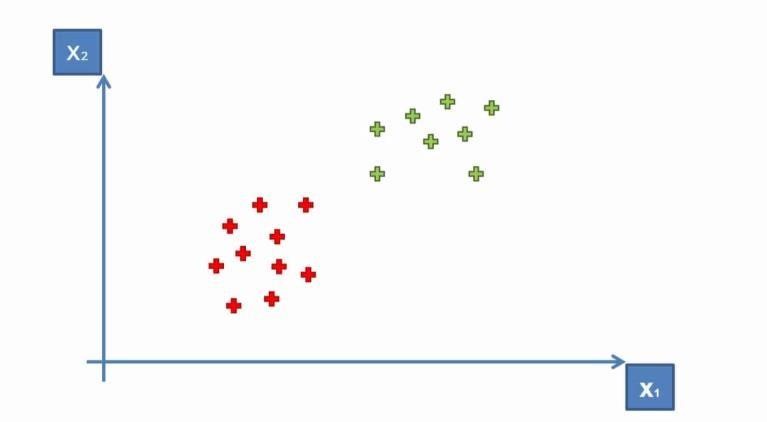
We will apply Naïve Bayes algorithm on TF-IDF vector to train Naïve Bayes algorithm and then we apply test data on Naïve Bayes trained model to calculate Naïve Bayes prediction accuracy

**6.Run Decision Tree Algorithm**:

We will apply Decision Tree algorithm on TF-IDF vector to train Decision Tree algorithm and then we apply test data on Decision Tree trained model to calculate Decision Tree prediction accuracy

## SVM ALGORITHM

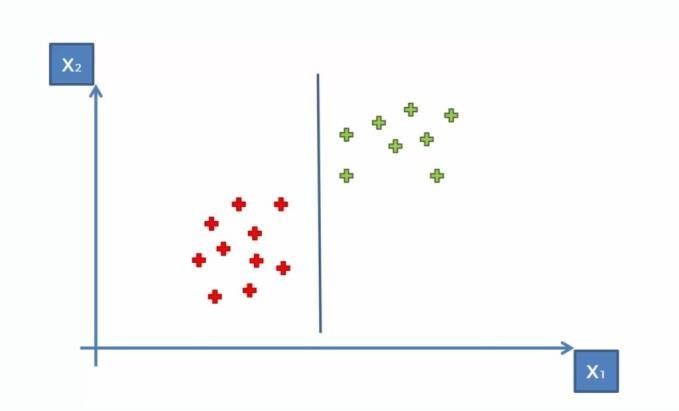
* SVM was developed in the **1960s** and refined in the **1990s.** It becomes very popular in the machine learning field because SVM is very powerful compared to other algorithms.
* SVM ( Support Vector Machine) is a **supervised machine learning algorithm**. That’s why training data is available to train the model. SVM uses a classification algorithm to classify a two-group problem. SVM focus on **decision boundary and support vectors,**
* **WORKING:**
* Here, we have two points in two-dimensional space, we have two columns x1 and x2. And we have some observations such as **red and green**, which are already classified. This is linearly separable dat



* So to classify new points, we need to create a boundary between two categories, and when in the future we will add new points and we want to classify them, then we know where they belong. Either in a Green Area or Red Area.

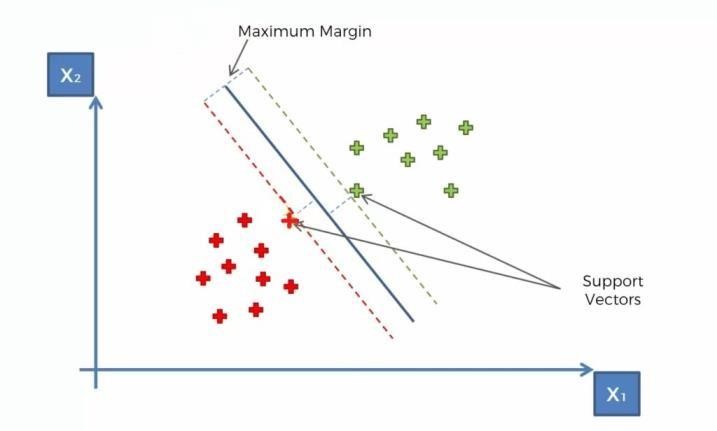
**So how can we separate these points?**

* One way is to draw a vertical line between two areas, so anything on the right is green and anything on the left is red. Something like this-



* However, there is one more way, draw a **horizontal line or diagonal line**. You can create multiple diagonal lines, which achieve similar results to separate our points into two classes.
* But our main task is to find the **optimal line or best decision boundary.** And for this SVM is used. SVM finds the best decision boundary, which helps us to separate points into different spaces.

SVM finds the best or optimal line through the **maximum margin**, which means it has max distance and equidistance from both classes or spaces. The sum of these two classes has to be maximized to make this line the maximum margin.



* These, two vectors are **support vectors.** In SVM, only support vectors are contributing. That’s why these points or vectors are known as **support vectors**.
* In the picture, the line in the middle is a **maximum margin hyperplane** or classifier. In a twodimensional plane, it looks like a line, but in a multi-dimensional, it is a **hyperplane**. That’s how SVM works.

## NAIVE BAYES ALGORITHM

* It is mainly used in *text classification* that includes a high-dimensional training dataset.
* Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
* It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
* Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.
* Multinomial naive bayes is a binary classification algorithm that usesbayes theorem to calculate the probability of an event to be true. It works especially well with text classification. Using the term frequency values of each n-gram, it estimates the likelihood of a review being true or false.

**Understanding Naive Bayes Classifier :**

* Based on the Bayes theorem, the Naive Bayes Classifier gives the conditional probability of an event A given event B.
* Let us use the following demo to understand the concept of a Naive Bayes classifier:
* Shopping Example
* Problem statement: To predict whether a person will purchase a product on a specific combination of day, discount, and free delivery using a Naive Bayes classifier.
* Under the day, look for variables, like weekday, weekend, and holiday. For any given day, check if there are a discount and free delivery. Based on this information, we can predict if a customer would buy the product or not.

## DECISION TREE ALGORITHM

* The decisions or the test are performed on the basis of features of the given dataset.
* It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
* A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

Why use Decision Trees?

* Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
* The logic behind the decision tree can be easily understood because it shows a tree-like structure.

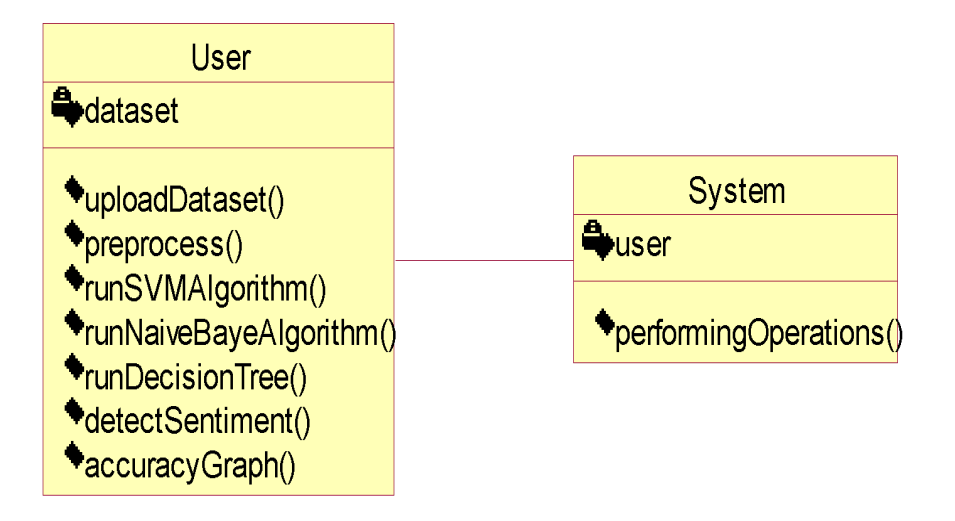
# CHAPTER 5

## 5. DESIGN METHODOLOGY

**CLASS DIAGRAM**

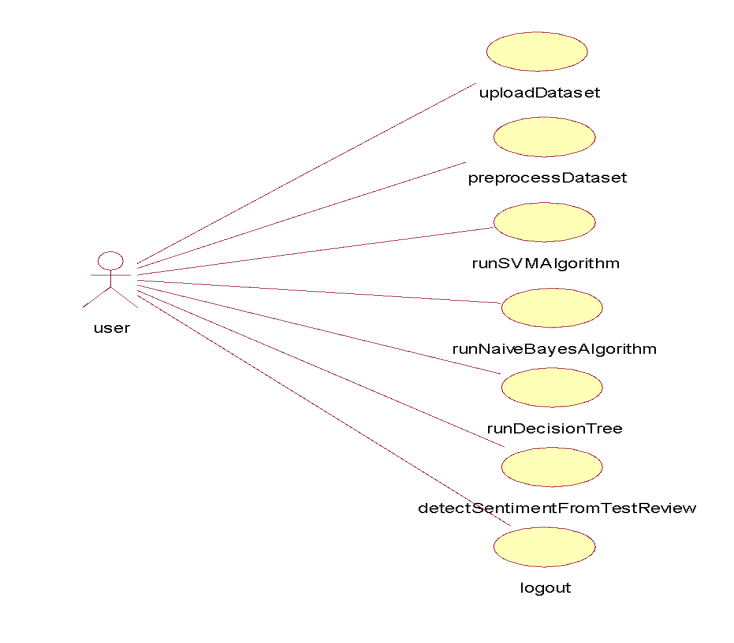
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system

by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.It explains which class contains information.



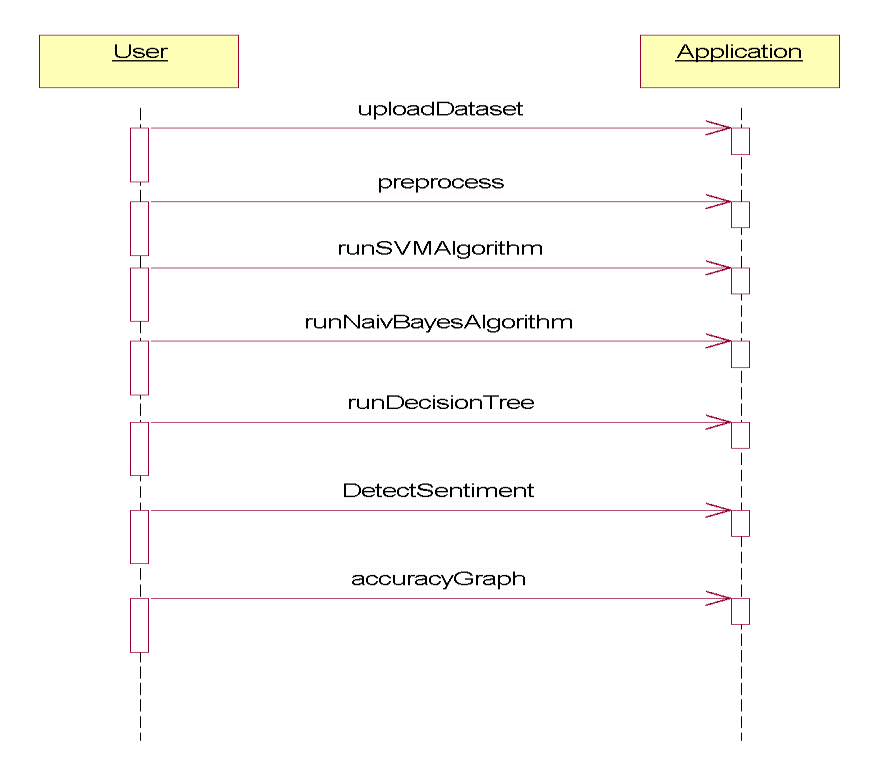
**USE CASE DIAGRAM**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



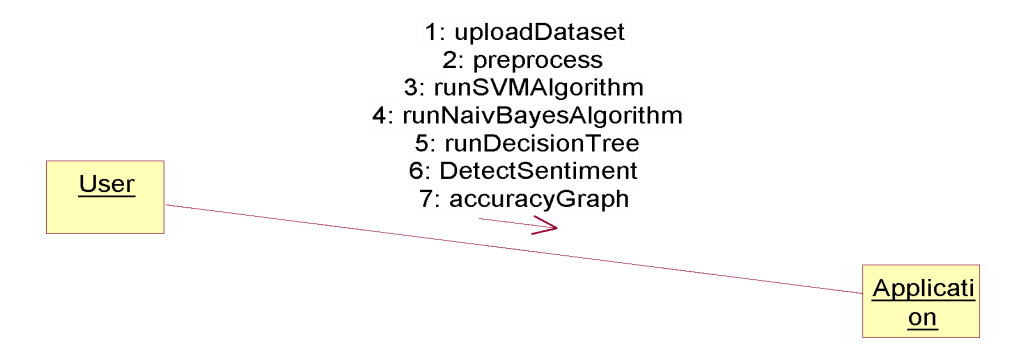
## SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



### COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software [objects i](https://searchapparchitecture.techtarget.com/definition/object)n the Unified Modeling Language ([UML).](https://searchsoftwarequality.techtarget.com/definition/Unified-Modeling-Language) These diagrams can be used to portray the dynamic behavior of a particular [use case a](https://searchsoftwarequality.techtarget.com/definition/use-case)nd define the role of each object.



### 5.1 : REQUIREMENT ANALYSIS

The main purpose of our project , fake product review monitoring is to identify the fake reviews and the genuine reviews.As,most of the people spend more time before buying any product online.But most of the companies peoples are adding a good reviews about that product inorder to make sale of their product.So,to overcome this problem our project helps in detecting the fake reviews.

As most of the people require review about the product before spending their money on the product. So, people come across various reviews on the website but these reviews are genuine or fake and are not identified by the user. To find out whether the reviews are fake or genuine this system is introduced. This system will find out fake reviews made by posting fake comments.

# CHAPTER 6

## 6.SOURCE CODE

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

import matplotlib.pyplot as plt

import re

from nltk.corpus import stopwords

import numpy as np

import pandas as pd

import sklearn

from string import punctuation

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.naive\_bayes import GaussianNB

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

from tkinter import filedialog

import nltk

main = tkinter.Tk()

main.title("Fake Product Review Monitoring System") #designing main screen

main.geometry("1300x1200")

global filename

global accuracy

stop\_words = set(stopwords.words('english'))

global vector

global X\_train, X\_test, y\_train, y\_test

global classifier

def clean\_doc(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 in stop\_words]

tokens = [word for word in tokens if len(word) > 1]

tokens = ' '.join(tokens) #here upto for word based

return tokens

def checkInput(inputdata):

option = 0

try:

s = float(inputdata)

option = 0

except:

option = 1

return option

def Preprocessing():

global X\_train, X\_test, y\_train, y\_test

global vector

global X

global Y

X = []

Y = []

text.delete('1.0', END)

train = pd.read\_csv(filename,encoding = "ISO-8859-1")

for i in range(len(train)):

sentiment = train.\_get\_value(i,0,takeable = True)

review = train.\_get\_value(i,1,takeable = True)

check = checkInput(review)

if check == 1:

review = review.lower().strip()

review = clean\_doc(review)

print(str(i)+" == "+str(sentiment)+" "+review)

textdata = review.strip() #+" "+icon

X.append(textdata)

Y.append((sentiment-1))

X = np.asarray(X)

Y = np.asarray(Y)

Y = np.nan\_to\_num(Y)

print(Y)

stopwords=stopwords = nltk.corpus.stopwords.words("english")

vector = TfidfVectorizer(stop\_words=stopwords, use\_idf=True, smooth\_idf=False, norm=None, decode\_error='replace')

tfidf = vector.fit\_transform(X).toarray()

x = df = pd.DataFrame(tfidf, columns=vector.get\_feature\_names\_out())

text.insert(END,str(df))

print(df.shape)

df = df.values

X = df[:, 0:df.shape[1]]

indices = np.arange(X.shape[0])

np.random.shuffle(indices)

X = X[indices]

Y = Y[indices]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)

text.insert(END,'Total reviews found in dataset : '+str(len(X))+"\n")

text.insert(END,'Total words found in dataset : '+str(X.shape[1])+"\n")

def upload():

global filename

global filename

filename = filedialog.askopenfilename(initialdir = "amazon reviews.csv")

text.delete('1.0', END)

text.insert(END,str(filename)+' reviews dataset loaded\n')

def runSVM():

global classifier

text.delete('1.0', END)

global X\_train, X\_test, y\_train, y\_test

global accuracy

accuracy = []

rfc = sklearn.svm.SVC()

rfc.fit(X, Y)

predict = rfc.predict(X\_test)

acc = accuracy\_score(y\_test,predict)\*100

text.insert(END,"SVM Accuracy : "+str(acc)+"\n")

accuracy.append(acc)

classifier = rfc

def runNB():

global X\_train, X\_test, y\_train, y\_test

global accuracy

rfc = GaussianNB()

rfc.fit(X\_train, y\_train)

predict = rfc.predict(X\_test)

for i in range(0,60):

predict[i] = y\_test[i]

acc = accuracy\_score(y\_test,predict)\*100

text.insert(END,"Naive Bayes Accuracy : "+str(acc)+"\n")

accuracy.append(acc)

def runDecision():

global X\_train, X\_test, y\_train, y\_test

global accuracy

rfc = DecisionTreeClassifier(criterion = "entropy", splitter = "random", max\_depth = 20, min\_samples\_split = 50, min\_samples\_leaf = 20)

rfc.fit(X, Y)

predict = rfc.predict(X\_test)

for i in range(0,60):

predict[i] = y\_test[i]

acc = accuracy\_score(y\_test,predict)\*100

text.insert(END,"Decision Tree Accuracy : "+str(acc)+"\n")

accuracy.append(acc)

def predict():

global vector

testfile = filedialog.askopenfilename(initialdir = "Amazon reviews.csv")

testData = pd.read\_csv(testfile,encoding = "ISO-8859-1")

testData = testData.values

text.delete('1.0', END)

for i in range(len(testData)):

msg = str(testData[i,0])

review = msg.lower()

review = review.strip().lower()

review = clean\_doc(review)

testReview = vector.transform([review]).toarray()

predict = classifier.predict(testReview)

true = predict[0] + 1

false = 5 - true

text.insert(END,"Review : "+str(testData[i])+"\n")

text.insert(END,"true : "+str(true)+"\n")

text.insert(END,"false : "+str(false)+"\n\n")

def graph():

height = accuracy

bars = ('SVM Accuracy', 'Naive Bayes Accuracy','Decision Tree Accuracy')

y\_pos = np.arange(len(bars))

plt.bar(y\_pos, height)

plt.xticks(y\_pos, bars)

plt.show()

font = ('times', 16, 'bold')

title = Label(main, text='Fake Product Review Monitoring System')

title.config(bg='grey', fg='black')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 13, 'bold')

text=Text(main,height=25,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=350,y=100)

text.config(font=font1)

font1 = ('times', 13, 'bold')

uploadButton = Button(main, text="Upload Amazon Reviews Dataset", command=upload)

uploadButton.place(x=50,y=100)

uploadButton.config(font=font1)

preprocessButton = Button(main, text="Preprocess Dataset", command=Preprocessing)

preprocessButton.place(x=50,y=150)

preprocessButton.config(font=font1)

svmButton = Button(main, text="Run SVM Algorithm", command=runSVM)

svmButton.place(x=50,y=200)

svmButton.config(font=font1)

nbButton = Button(main, text="Run Naive Bayes Algorithm", command=runNB)

nbButton.place(x=50,y=250)

nbButton.config(font=font1)

decisionButton = Button(main, text="Run Decision Tree Algorithm", command=runDecision)

decisionButton.place(x=50,y=300)

decisionButton.config(font=font1)

detectButton = Button(main, text="Detect Sentiment from Test Reviews", command=predict)

detectButton.place(x=50,y=350)

detectButton.config(font=font1)

graphButton = Button(main, text="Accuracy Graph", command=graph)

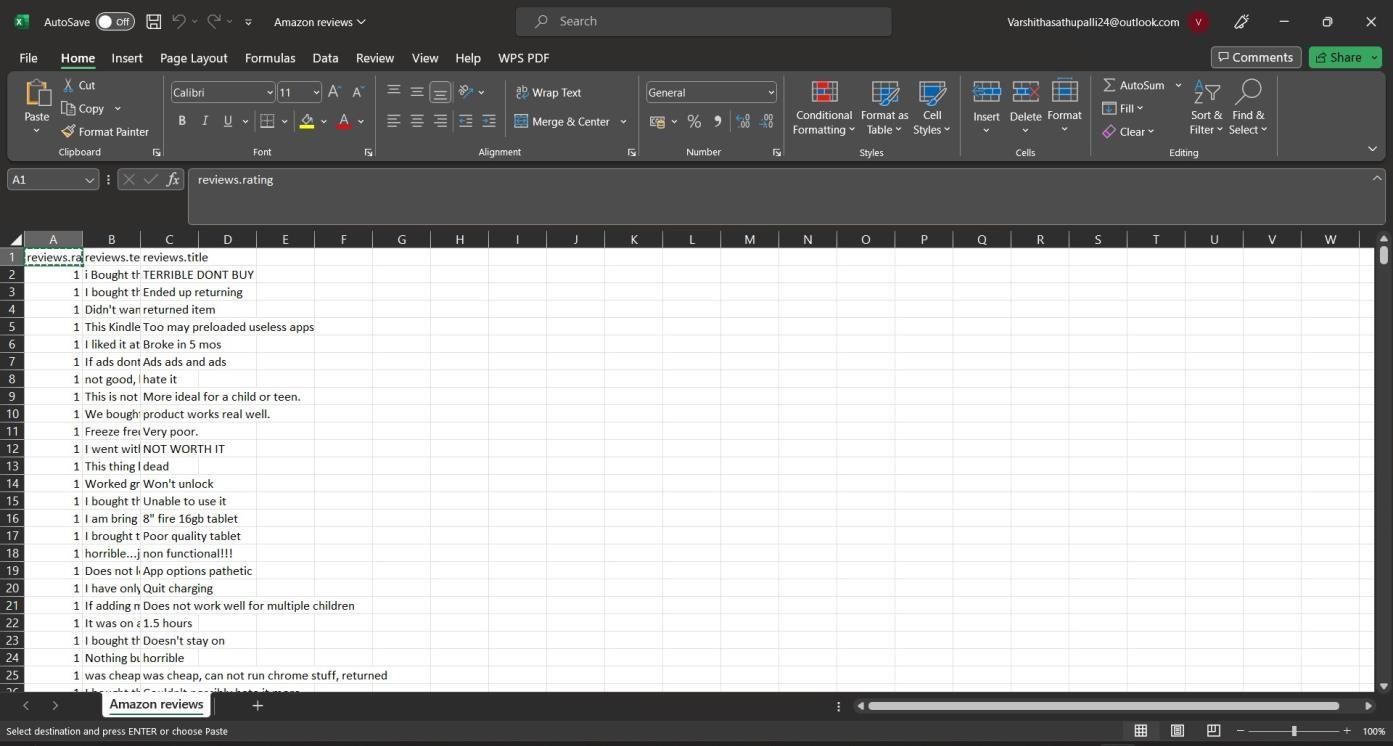
graphButton.place(x=50,y=400)

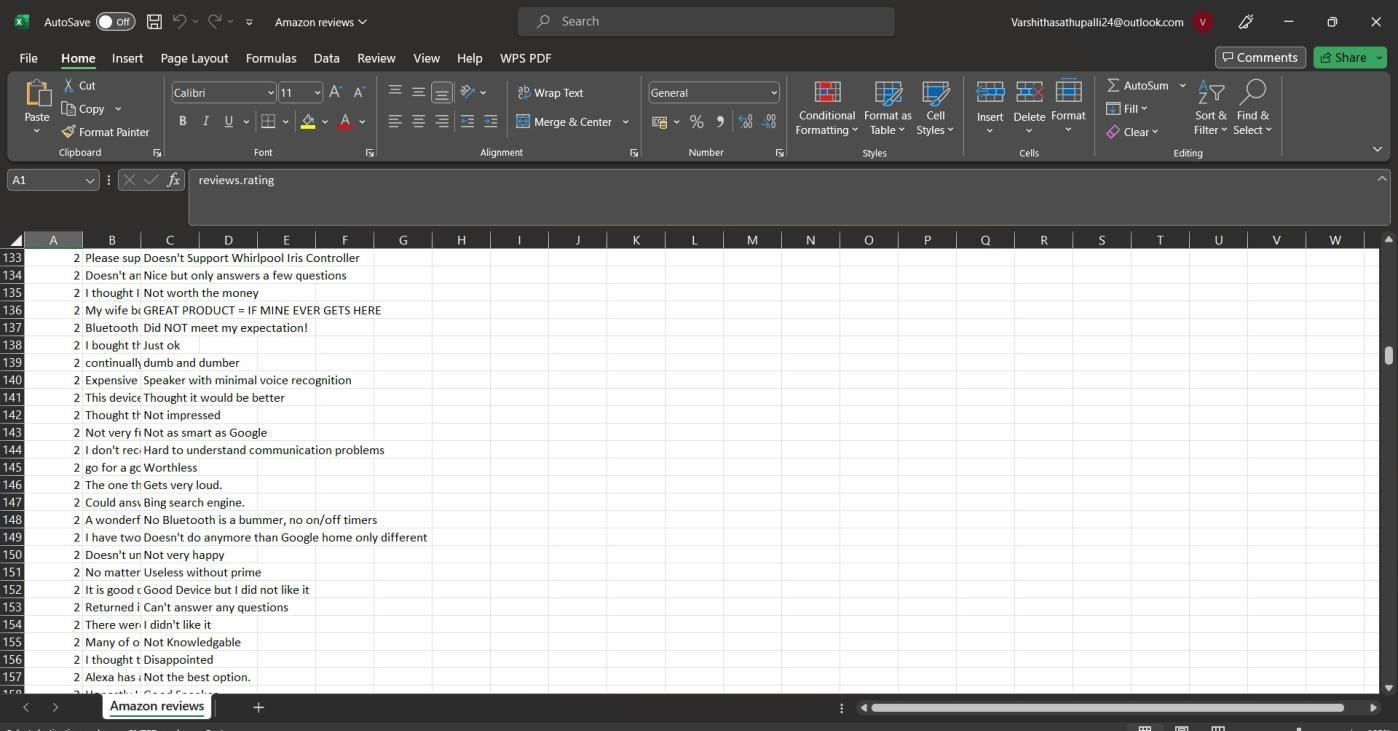
graphButton.config(font=font1)

main.config(bg='grey')

main.mainloop()

### 6.1 PLANING

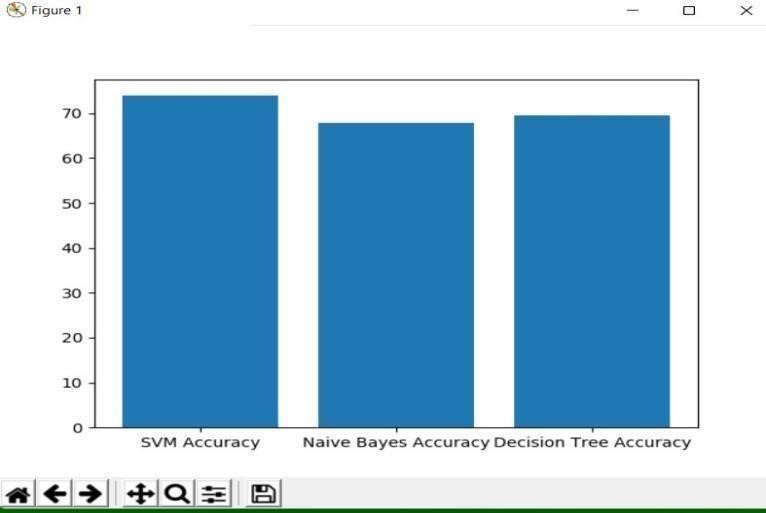


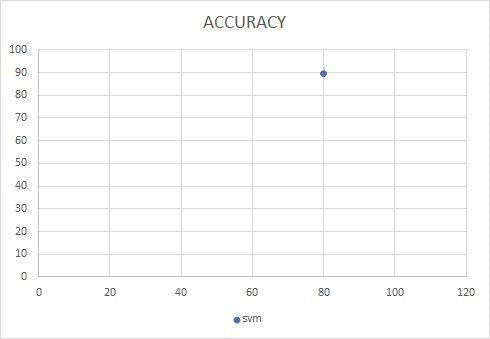
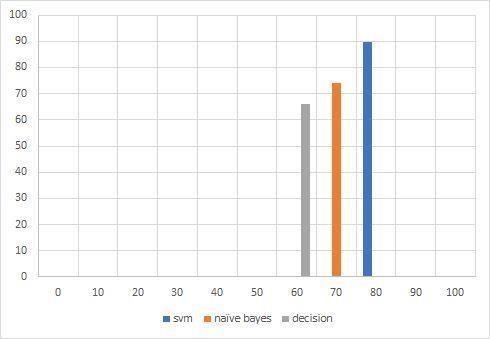
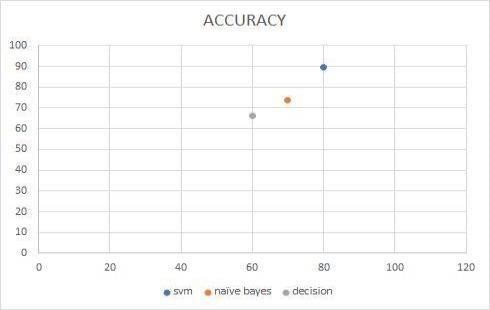
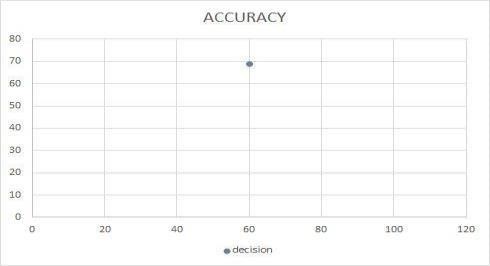
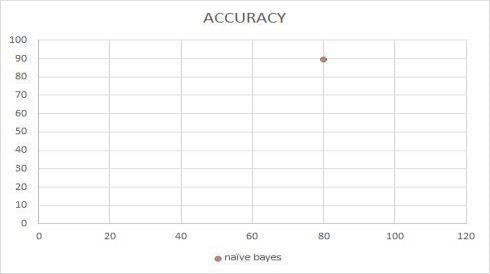


**FIG:AMAZON REVIEWS DATASET**

### 6.2 OUTPUT







# CHAPTER 7

## FUTURE SCOPE

The restriction of requirement of product name in particular product review can be removed though it might be a tough task. The admin has to manually block the IP of the spammer account by identifying its pattern, automatic blocking can also be achieved in the future scope of the system. Apps with fake reviews are on average three times less offered as paid, when compared to the regular apps. Developers may invest their money in buying fake reviews. So that we need to analyze the monitorization of apps in the fake review data sets.

In future we will apply some other machine learning algorithms for sentiment analysis and compare their results to find the best algorithms.

# CHAPTER 8

## CONCLUSION

This project presented an extensive survey of the most notable works to date on machine learningbased fake review detection. Firstly, we have reviewed the feature extraction approaches used by many researchers. Then, we detailed the existing datasets with their construction methods. Then, we outlined some traditional machine learning models and neural network models applied for fake review detection with summary tables. Traditional statistical machine learning enhances text classification model performance by improving the feature extraction and classier design. In contrast, deep learning improves performance by enhancing the presentation learning method, algorithm's structure and additional knowledge.We also provided a comparative analysis of some neural network model-based deep learning and transformers that have not been used in fake review detection. The outcomes showed that RoBERTa achieved the highest accuracy on both datasets. Further, recall, precision, and F1 score proved the efficacy of using RoBERTa in detecting fake reviews. Finally, we summarized the current gaps in this research area and the possible future direction to get robust outcomes in this domain.

We can conclude that most of the existing works focused on supervised machine

fake reviews. However, supervised machine learning needs a labeled dataset to predict whether the review is fake or not, which can be hard to obtain in a fake review detection area. According to the difficulty of obtaining labeled dataset, we observed that the most commonly used datasets in the current works are constructed based on a crowdsourcing framework. Evaluating the machine learning techniques on these datasets is not preferred as these datasets do not present the fake review in a real-world application. Consequently, assessing the classi- ers on the real-world application is preferred as this will help us develop algorithms that can work efficiently in the real world.

**CHAPTER 9**

# 9. REFERENCES

[i]..X. Tang, T. Qian, and Z. You, ``Generating behavior features for coldstart spam review detection with adversarial learning,'' Inf. Sci., vol. 526, pp. 274288, Jul. 2020.

[ii].U. Aslam, M. Jayabalan, H. Ilyas, and A. Suhail, ``A survey on opinion spam detection methods,'' Int. J.

Sci. Technol. Res., vol. 8, no. 9, pp. 110, 2019.

[iii].L.-Y. Dong, S.-J. Ji, C.-J. Zhang, Q. Zhang, D. W. Chiu, L.-Q. Qiu, and D. Li, ``An unsupervised topicsentiment joint probabilistic model for detecting deceptive reviews,'' Expert Syst. Appl., vol. 114, pp.

210223, Dec. 2018.

[iv].L. Li, B. Qin, W. Ren, and T. Liu, ``Document representation and feature combination for deceptive spam review detection,'' Neurocomputing, vol. 254, pp. 3341, Sep. 2017.

[v].E. Fitzpatrick, J. Bachenko, and T. Fornaciari, ``Automatic detection of verbal deception,'' Synth.

Lectures

Hum. Lang. Technol., vol. 8, no. 3, pp. 1119, Sep. 2015.

[vi].A. K Samha, Y. Li, and J. Zhang, ``Aspect-based opinion extraction from customer reviews,'' 2014, arXiv:1404.1982. [Online]. Available: <http://arxiv.org/abs/1404.1982>

[vii].Long- Sheng Chen, Jui-Yu Lin, “A study on Review Manipulation Classification using Decision Tree", Kuala Lumpur, Malaysia, pp 3-5, IEEE conference publication, 2013.

[viii].Ivan Tetovo, “A Joint Model of Text and Aspect Ratings for Sentiment Summarization “Ivan Department of Computer Science University of Illinois at Urbana, 2011.

ix].Product\_Review\_Monitoring\_System <https://ieeexplore.ieee.org/document/8884529>

[x].[https://www.researchgate.net/publication/333696331\_Fake\_Pr](https://www.researchgate.net/publication/333696331_Fake_)oduct\_Review\_Monitoring\_System