Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon

College of information Technology
Department of Information Networks

LEVERAGING SOCIAL MEDIA NETWORKS FOR HATE SPEECH CLASSIFICATION

A Thesis
Submitted to the Council of the College of Information Technology at University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology / Information Networks

By
Laith abbas Abdullah

Supervised by
Asst. Prof.

Dr. Mehdi Ebady Mana

٢٠٢٠A.D

١٤٤١A.H

□

(فَتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ وَلَا
تَعْجَلْ بِالْقُرْآنِ مِنْ قَبْلِ أَنْ
يُقْضَى إِلَيْكَ وَحْيُهُ وَقُلْ رَبِّ
زِدْنِي عِلْمًا)

صدق الله العظيم
سورة طه –آية (١١٤)

Supervisor Certification

I certify that this thesis was prepared under my supervision at the Department of Information Networks / Collage of Information Technology /University of Babylon, by **laith abbas Abdallah** as a partial fulfillment of the requirements for the degree of **Master in Information Technology**.

Signature:

Name: **Dr. Mehdi Ebady Mana**

Title: **Assistant Professor**

Date: / / 2020

The Head of the Department Certification

In view of the available recommendation, we forward this thesis for debate by the examining committee.

Signature:

Name: **Prof. Dr. Saad Talib Hasson**

Title: **Professor**

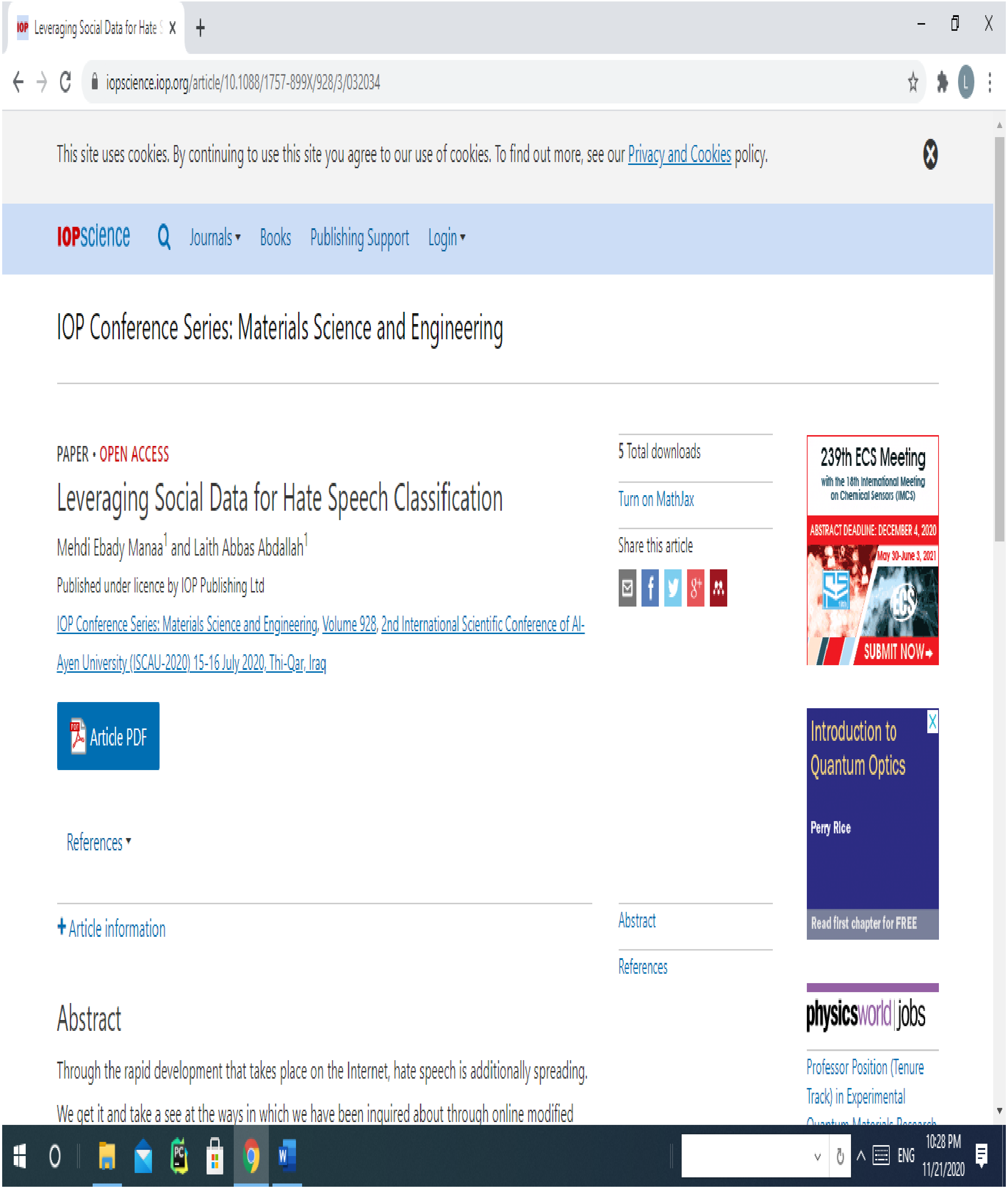
Date: / / 2020

I hereby declare that this thesis, submitted to the University of Babylon in partial fulfillment of the requirement for the degree of Master in Information Technology \ Information Networks, has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for experts and summaries whose source is appropriately cited in the references.

Signature:

Name: **laith Abbas Abdullah**

Date: / / 2020



Leveraging Social Data for Hate S X
+
iopscience.iop.org/article/10.1088/1757-899X/928/3/032034

IOPscience
Journals
Books
Publishing Support
Login

IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

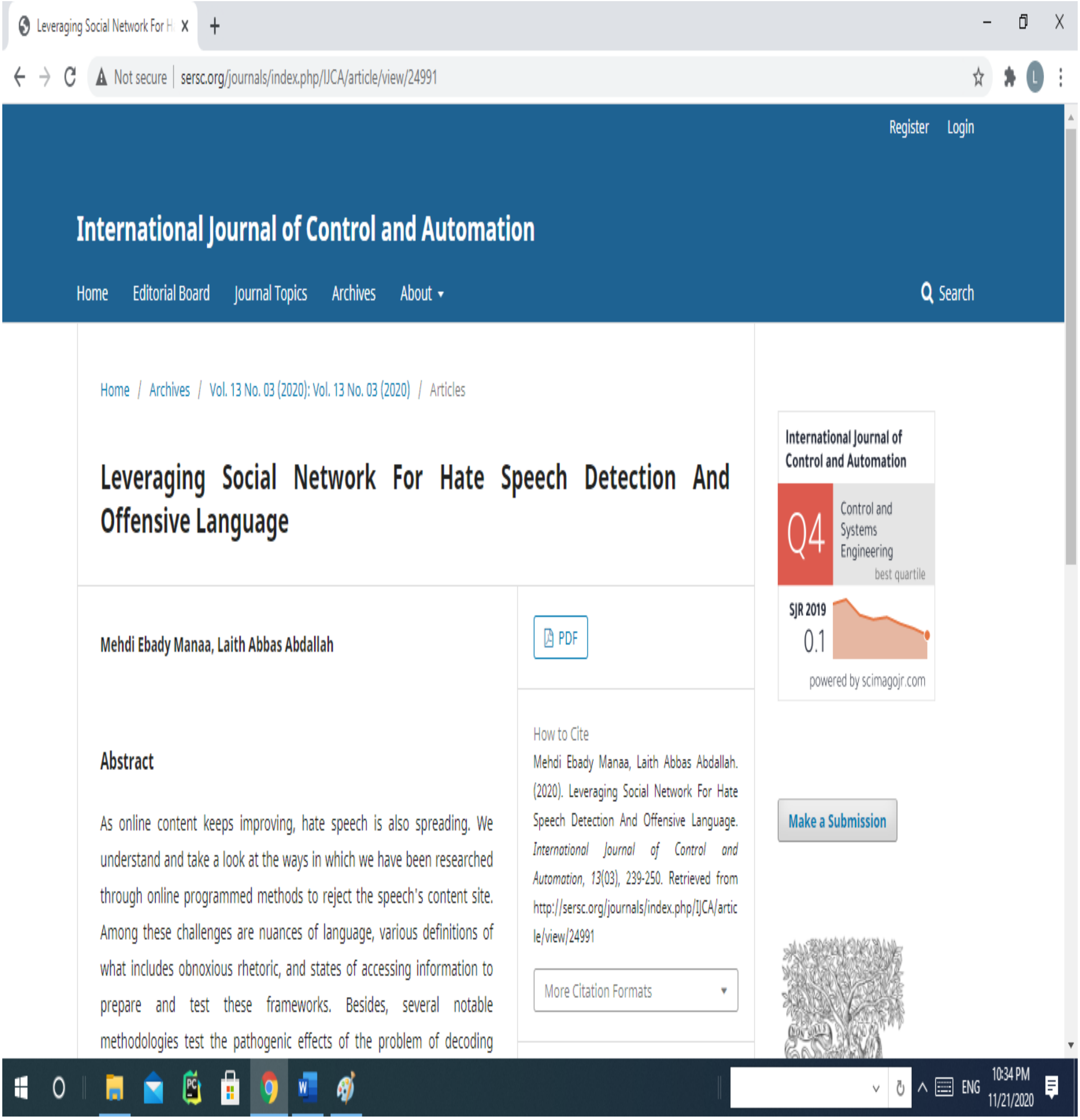
Leveraging Social Data for Hate Speech Classification

Mehdi Ebady Manaa[1] and Laith Abbas Abdallah[1]

Published under licence by IOP Publishing Ltd
IOP Conference Series: Materials Science and Engineering, Volume 928, 2nd International Scientific Conference of Al-Ayen University (ISCAU-2020) 15-16 July 2020, Thi-Qar, Iraq

Abstract

Through the rapid development that takes place on the Internet, hate speech is additionally spreading. We get it and take a see at the ways in which we have been inquired about through online modified



International Journal of Control and Automation

Home / Archives / Vol. 13 No. 03 (2020): Vol. 13 No. 03 (2020) / Articles

Leveraging Social Network For Hate Speech Detection And Offensive Language

Mehdi Ebady Manaa, Laith Abbas Abdallah

Abstract

As online content keeps improving, hate speech is also spreading. We understand and take a look at the ways in which we have been researched through online programmed methods to reject the speech's content site. Among these challenges are nuances of language, various definitions of what includes obnoxious rhetoric, and states of accessing information to prepare and test these frameworks. Besides, several notable methodologies test the pathogenic effects of the problem of decoding

How to Cite

Mehdi Ebady Manaa, Laith Abbas Abdallah. (2020). Leveraging Social Network For Hate Speech Detection And Offensive Language. *International Journal of Control and Automation*, *13*(03), 239-250. Retrieved from http://sersc.org/journals/index.php/IJCA/article/view/24991

ABSTRACT

Developments in online communication and content have been accompanied with the spread of hate speech, which has become a major issue of concern to researchers in a number of overlapping fields.

Hate crimes are overt acts that can include acts of violence against persons or property, violation or deprivation of civil rights, certain "true threats," or acts of intimidation, or conspiracy to commit these crimes. There are many problems that face the researcher in classifying hate speech, including the Subtleties of vocabulary, conflicting meanings of what constitutes hate speech and limits of the availability of data for training and testing. One of the main challenges that are faced is the accurate classification of hate speech in online social media platforms, and it will therefore be addressed in this thesis.

The main aims of the study are to use data mining techniques in classification hate speech within large web documents, and to implement a supervised classification analysis of the obtained data. This is followed by evaluating the proposed method in light of alternative implementation techniques.

The proposed system uses n-grams (k-shingles) with the Minhash algorithm and term frequency-inverse document frequency, and the document to vector model with a number of machine learning algorithms (Naive Bayes, Support Vector Machine, random forest, Logistic Regression and Neural Network). Proposed system applied to four type

Table of Contents

List OF Figures

List of Abbreviations

LRC	Logistic Regression classifier
TF-IDF	Term Frequency and Inverse Document Frequency
RF	Random forest classifier

NLP	Neuro-linguistic programming
PR	Precision
SVM	Support Vector Machine
AC	Accuracy
D2V	Document to vector
W2V	Word to vector
PV-DM	Distributed Memory Architecture of Paragraph Vectors
SGD	Stochastic gradient descent
OSN	Online social network
NSA	National-Security Agency
RDF	Resource Description Framework
MD5	Message Digest 5
SHA	Secure Hash Algorithm
CRC	cyclic redundancy check
PV-DBOW	Bag of Words version of Paragraph Vector
NB	Naive Bayes
NLTK	Natural Language Tool Kit

الاستفادة من شبكات وسائل التواصل الاجتماعي لتصنيف خطاب الكراهية

نبذة مختصرة

ترافقت التطورات في الاتصالات والمحتوى عبر الإنترنت مع انتشار خطاب الكراهية، الذي أصبح قضية رئيسية تشغل بال الباحثين في عدد من المجالات المتداخلة. جرائم الكراهية هي أعمال علنية يمكن أن تشمل أعمال عنف ضد الأشخاص أو الممتلكات، أو انتهاك أو حرمان من الحقوق المدنية، أو بعض "التهديدات الحقيقية"، أو أعمال التخويف، أو التآمر لارتكاب هذه الجرائم. هناك العديد من المشكلات التي تواجه الباحث في تصنيف خطاب الكراهية، بما في ذلك التفاصيل الدقيقة للمفردات، والمعاني المتضاربة لما يشكل خطاب الكراهية، وحدود توافر البيانات للتدريب والاختبار. أحد التحديات الرئيسية التي نواجها هو التصنيف الدقيق لخطاب الكراهية في منصات التواصل الاجتماعي عبر الإنترنت، وبالتالي سيتم تناوله في هذه الأطروحة. تتمثل الأهداف الرئيسية للدراسة في استخدام تقنيات التنقيب عن البيانات في تصنيف خطاب الكراهية ضمن مستندات الويب الكبيرة، وتنفيذ تحليل تصنيف خاضع للإشراف للبيانات التي تم الحصول عليها. ويلي ذلك تقييم الطريقة المقترحة في ضوء تقنيات التنفيذ البديلة. يستخدم النظام المقترح n-grams (k-shingles مع خوارزمية Minhash ومصطلح التردد العكسي للمستند (TFIDF) ، ونموذج Doc2Vec مع عدد من خوارزميات التعلم الآلي (Naive Bayes و Support Vector Machine (SVM) والشبكة العصبية). وأظهرت النتائج التي تم الحصول عليها أن نسبة الدقة في تحديد الوظائف (٩٠-٩٩) ٪ ضمن أربع فئات. يتفوق النهج المقترح على جميع الأساليب الحديثة مع زيادة كبيرة في الدقة.

ABSTRACT

Developments in online communication and content have been accompanied with the spread of hate speech, which has become a major issue of concern to researchers in a number of overlapping fields. Hate crimes are overt acts that can include acts of violence against persons or property, violation or deprivation of civil rights, certain "true threats," or acts of intimidation, or conspiracy to commit these crimes. There are many problems that face the researcher in classifying hate speech, including them Subtleties of vocabulary, conflicting meanings of what constitutes hate speech and limits of the availability of data for training and testing. One of the main challenges that are faced is the accurate classification hate speech in online social media platforms, and it will therefore be addressed in this thesis. The main aims of the study are to use data mining techniques in classification hate speech within large web documents, and to implement a supervised classification analysis of the obtained data. This is followed by evaluating the proposed method in light of alternative implementation techniques. The proposed system uses n-grams (k-shingles) with the Minhash algorithm and Document Reverse Frequency (TFIDF) term, and the Doc2Vec model with a number of machine learning algorithms (Naive Bayes, Support Vector Machine (SVM), and Neural Network). The results obtained indicated an accuracy rate of (90-99) % in identifying posts within four categories. The proposed approach outperforms all modern methods with a significant increase in accuracy.

Chapter One

Introduction

1.1 Introduction

Data analysis is considered to be one of the hot topics that are used in many applications such as education, business, econometrics and community data, with the main focus being on the data of social issues, found on social media platforms like Twitter, Facebook, etc. The hate speech has increased in recent years as a result of different criteria and rules for classifying speech that is included in hate speech on different social media platforms [1].

In fact, hate and violence crimes are not new, yet social media is one of the many outlets that enable the spread of these crimes. For example, there are some people suspected of a variety of hate speech attacks related to media records and electronic posts on social media [2].

Among the many interactions observed on social media, there are some video recordings of people who are suspected of the 2019 terrorist attack in New Zealand as the latter led to the death of 49 people who were shot on live streaming. It was broadcasted on one of the most important social media platforms, namely Facebook, and it was quickly reposted onto different social media like Twitter, YouTube, and Instagram many huge numbers of people have been duplicating and sharing the footage online, many others responded with disgust - urging others not only not to share the footage, but not even to watch it.spreading the video, many said, was what the attacker had wanted people to do. [3].

What distinguishes electronic communication forums, especially social media, is the possibility of expressing opinions and ideas freely about various issues. However, when expressing one's opinion freely, the privacy of the individual is the most important right which must be respected, thereby leaving everything that might encourage spreading hatred and violate a person’s right to these forums. It is worth noting that in some countries, hate speech is not considered to be illegal behavior unless it leads to any form of physical violence, as is the case in the United States [4].

Social media is considered to be one of the most popular methods of communication, including social forums commonly used nowadays like Facebook, YouTube, and Twitter. Therefore, it is important to restrict those forms of speech which increase hated and hurt other users on the platform by means of a special policy followed by these forums to eliminate hatred in dialogue and publications of individual behavior [5].

Hate speech was and remains an active field of academic research, especially for the sociology community. It is found to be one of the problems that cannot be solved easily in most societies, especially when it involves topics that include racism between black and white people, where hate speech causes huge problems and issues among people. Because of the many undesirable social consequences and cases of repression, policies have pushed for a specific way to organize hate speech [6].

Being a controversial topic, online hate-speech is characterized by persistence and the possibility of a return, as the hate speech can be active for a long time or be activated at any time. It also has an anonymity feature, where people on social media share content without showing their true identity. Hence, it is possible to escape punishment and the law for their belief in not revealing their identity during hate-speech. There are many definitions that relate to the concept of hate-speech in various social media platforms. Table (1.1) presents this concept as defined by the most comprehensive and commonly used online platforms, namely Facebook, Twitter, and YouTube [7].

Table 1.1: Definition of hate speech as stated by Facebook, Twitter, and YouTube.

Source	Definition
Facebook	Hate speech has been defined as a targeted attack on a person's protected characteristics of race, sexual orientation,

	personal identity, religious affiliation, and serious illnesses.
Twitter	It is not permissible to promote users violence against people and their characteristics distinguished by race, national origin and sexual orientation, age, religious affiliation, and serious illness, as well as accounts that incite hate speech according to these categories, the ways to punish a violation of these rules differ depending on the severity of the violation. The penalties range from asking the person to remove his/her irregular tweets, to the eventual suspension of the account.
YouTube	Freedom of expression is encouraged, as they try to defend the right of people to express their views, but hate speech is not allowed. An expression is considered hate speech directly if it concerned specific features such as ethnicity, disability, age, gender, and religion, as well as military status and identity of the person. There is a fine line that distinguishes between what is considered hate speech and what is not. In general, criticism is acceptable, but if hatred is incited against people because of their race, religion, or other basic features, it is considered a violation of our policy.

1.2 Related works

There are many research trends related to hate speech, which is relevant to our topic, as there are trends in hate speech detection. Including what concerns the resulting accounts, author profiling and others which including datasets in social media data.

A specific method for assessing machine learning techniques on Twitter to reveal hate speech in South Africa has been relied in [8]. It has witnessed many recent ethnic attacks on Twitter. the used system developed a specific away based on the range of English domains for tweets from South Africa they have been used character n-gram, word n-gram, negative sentiment, and syntactic-properties based on the special executable. Where the results showed the use of character n-gram is the best way to uncover hate speech compared with word n-gram, as optimized support vector machine with character n-gram performed in detection of hate speech with true positive rate of

89.4%, while optimized gradient boosting with word n-gram performed in detection of hate speech with true positive rate of 86.7%

In [9] the authors, propose an approach for hate speech detection against politicians in Arabic, they use SVM and NB to distinguish between the comments/messages containing hateful from the other, the purpose behind these experiments is to highlight the best technique for corpus construction and to compare among the different machine learning (Linear SVC, logistic regression(LR), Stochastic gradient descent (SGD) and CNN, LSTM, etc.) the best results concerning hate speech detection (up to87% accuracy) are achieved using LSVC and Bi-LSTM classifier with SG model of Word2vector

A good approach has been proposed in [10]. It solved problems related to previous research directions that are inattentive to user and community information and which are entirely dependent on texts and their processing. They suggested a model that incorporates Twitter-based profiling features. The data collected from Twitter as 16k tweets which explained racism and sexism, they implemented the Keras, Theano back-end and apply 200-dimensional pre-trained GLoVe word embeddings and Lightgbm models. These models have shown results that significantly outperform current trends of hate speech by adopting a qualitative analysis of certain characteristics

A method that uses community interaction to determine the content of hate speech on social media via a form known as Arabic Religious Hate Speech Net (ARHNet) used in [11] to expose the religious hate speech by including Arabic words and graphs. They illustrated a set of examples in Arabic that incite hatred among Muslims and the Jews, the methods have been used as Follower and Retweet graphs. They obtained a highest f1-score of 0.78 as com-pared to 0.77 through using Long short-term memory (LSTM), convolutional neural network (CNN) and Cancer Intervention and Surveillance Modeling Network (CISNet) model

In [12] the authors, relied on discovering hidden topics between texts related to cases of domestic violence on Twitter where they collected a large number of messages under the term domestic violence where they used the method of Machine learning methodology as a Latent Dirichlet allocation. In addition to relying on technologies Data mining techniques to generate a set of words within the proposed database rules and represented as popular bigrams as pairs and words of data sets. They collected 322,863 Tweets as document population. Among all collected Tweets, there are 80,868 bigrams (e.g., "domestic violence,”” stop domestic”). They choose the 20 most common words (16.72%) with the highest percentage in all 80,868 bigrams (100%)

In [13] the authors, started a speedy restraint to the new "German NetDG" law by allowing a model to identify "hate-speech" in the "German-language". They examined up to 50,000 right-wing hate tweets German posted between Aug-2017 and Apr-2018, at the period of the 2017 federal German elections. besides, it confirmed a global analysis of "qualitative/quantitative" within what shape of the hate speech from the political discourse view

In [14] the authors, A proposed study to analyze anti-Islam hate speech on social media networks as it is a growing concern for societies. They used an automated programming tool to distinguish between hate speech and Islamophobia. The results proved high accuracy of distinguishing this speech. Through the adoption of various types of classifications and the use of different algorithms to examine performance such as the Naïve-Bayes, Random Forests, Logistic Regression, and so on. The accuracy is 77.6% and balanced accuracy is 83%. The used tool enables future quantitative research into the drivers, spread, prevalence and effects of Islamophobic hate speech on social media

A deep neural multi-modal approach has been proposed in [15] to expose hate speech within the semantic meaning of texts in a social and cultural environment. In addition, the suggested methodology providing scalable insights with a comprehensive procedure for assessing various modeling techniques. They use techniques to deal with extracting features of semantic meaning for cultural, social contexts, while the evaluation based on traditional models and other baselines deep learning models. The importance of the proposed model emerged with the discovery of social groups that incite hate speech.

In [16] the authors, examined linking convolutional and got recurrent unit networks, and analyzed the pattern execution against all earlier deep learning paradigms. authors declared the work produces a new benchmark for future studies area. They captured both word sequence and order information in short texts, and it sets new benchmark by outperforming on 6 out of 7 datasets by between 1 and 13% in F1.

While, in [17] the authors, maintained that deep neural networks have a huge potential to determine the effect of hate speech exposure. authors used a deep learning method outperformed all the state-of-art proposals. The results have confirmed the original hypothesis of improving the classifier’s performance by employing additional user-based features into the prediction mechanism, the result achieved 93.2 % F-score.

In [18] the authors, suggested a specific framework that depended on data mining techniques to reveal the extreme content in the public Facebook posts. Where they adopted a method to automatically extract messages from public Facebook pages using API calls graph and filtering techniques to reveal extreme and violent content. A large database of data flow gets acquired from Facebook. Which helps law and rescue analysts know who perpetrated cybercrime. The results obtained from this kind of automatic translation service are rarely as good as if a human expert had translated the content of a website, but the great advantage with automatic translation is obviously the speed.

In [19] the authors, depending on the method of collecting people’s tweets on Twitter, there are several ways including the Hashtag graph for removing human-readable language junk, relating useful tweets, Text-preprocessing steps, and data standardization converting, and so on. They try to know the intensity and time of mass violence with data mining, which allows the authorities and the concerned authorities to address and prevent these cases. The final result is based on the number of tweets belonging to a particular area of the graph, the relation between the graphs is obtained through the fact that they have already provided a unique integer value to every tweet.

A survey research trends on automatic detection of text in hate speech has been made in [20]. It is also interested in discussing the difficulty of dealing with hate speech and provides a unified definition for the definition of hate speech in online social media and digital media platforms, in this stay they use N-grams, Part-of-speech (POS),rule-based approaches, sentiment analysis, and deep learning.

Open-source program that is used in the distributed environment has been proposed in [21] to discover hate speech, which is known as the Apache Spark. Where they developed this program, based on a special model for classification Amharic language for Facebook, through comments and posts, to hate talk, not hate, where used to learn a Random Forest and Naïve Bayes. While applying a special method to determine the characteristics depends on Word2Vec and the method of digital statistics that you know Frequency Term - Inverted Document Frequency (TFIDF). They tested by 10-fold cross-validation, the model based on word2vec embedding performed best with 79.83% Accuracy.

An automatic model has been proposed in [22], to classify tweets from a Twitter data set into three categories. The following category is hate, offensive, and clean speech. Where the experiments were examined using a feature n-grams dealing with the value of term frequency-inverse document frequency (TFIDF), which pass to more than multiple machines to learn. Besides, they creating a model that acts as an intermediary between the user and the Twitter platform. They based on tweets through Twitter REST API, the programming language used is Python, with the Tweepy library Twitter APIs, JSON document, and knowledge gathered from the features like the author of the tweet used for further analysis. The results showed that Logistic Regression performs better with the optimal n-gram range 1 to 3 for the L2 normalization of TFIDF. They achieved 95.6% accuracy upon evaluating it on test data.

An automatic data detection method within a labor-intensive task has been adopted in [23]. Where the risk resulting from the insulting statements is evaluated and the reference objectives are determined depending on the methods from them binary classification task to identify hostile hate phrases. The work exposes hostility towards strangers. Consequently, the results revealed that there were large amounts of abusive data and that many victims were detected correctly. As evaluation metrics were applied precision, recall and f1-measure. Compared to a machine learning baseline classifier. The pattern-based approach yields substantial precision values (75.26% and 73.89%) and moderate overall classification performance in terms of f1 value (67.91% and 62.03%).

Alarming diffusion prevention for a series of hate speeches campaigns within Facebook has been proposed in [24]. where the activity relied on the content of comments on general Italian pages. The work was divided into several phases, including distinguishing the type of hate, where a level was set for the number of repeated words, such as hate wood, up to five human comments, and also the integration of words in dictionaries, Where they relied on the algorithms of data analysis in machine-learning which represented as Support Vector Machines (SVM), and the second type which is considered an architectural of deep learning approach as LSTM. The results show the effectiveness of the two classification approaches tested over the first manually annotated Italian Hate Speech Corpus of social media text, SVM classifier reach accuracy 80.6% and LSTM 79.8%.

Investigating a user merits has been employed to improve the identification and bullying categorization and offensive Twitter user's activity in [25]. So, the robust extracting methodology used for text and attributes of the network-based features implementation through studying aggressors and bullies' properties such as the friends' number, followers exchanged, and deployment within the network, mainly helpful and operative within assorting attacking user activity. They using a corpus of 1.6M tweets posted over 3 months, and show that machine learning classification algorithms can accurately detect users exhibiting bullying and aggressive behavior, with over 90% Area Under Curve (AUC).

Providing a large-scale measurement proposed in [26] to study the main goals of hate speech in social media methods represented by Whisper and Twitter. In addition to developing and verifying a special methodology for identifying hate according to Datasets from the mentioned social media. The methodology based on classification of hate targets and examples of hate targets within the posts of Twitter and Whisper. The results show the used system identify the online hate speech and producing areas for prevention and disclosure strategies.

In [27] the authors, relied on predictive features to discover hate speech in one of the most popular social media like the Tweeter. It provided a list of criteria based on critical race theory to provide down a group of more than 16k tweets via analyzing the effect of linguistic features with the nature of letters n-grams as the indicative character of the n-gram, also, they provide a dictionary with the guiding words used in this work, in addition, the hate-speech was compared between different genders as the Racism, Sexism, and Neither, also other indicators like pronouns, honorifics, and gender-specific nouns. The most frequent words were clarified according to certain terms and also calculate the lengths of these items according to genders.

Adopted a distinctive method has been proposed in [30] for detecting hate-speech using vector machine classifier, n-grams trained words, and brown clusters approach. Where they collected samples of offensive speech and compare it with hate-speech. As well as the use of a mechanism to detect spelling errors to evade hate speech analysis. Where get benefits from using the concept of Yarowsky's algorithm as a template strategy to create other properties as corpus which is used to analyze the hate-speech. They find the smallest feature set has the best performance, reaching a precision of 68% with recall at 60%.

1.3 The problem statements

To address the hate speech problem, we proposed a solution to the detection of hate speech and offensive language on Twitter.

1.4 Research Aim and Objectives

The proposed work aims to detect hate speech in social media by applying Minhash technology, message analysis, and machine learning algorithms. The objectives can be divided into the following:

1. Finding hate speech within a large set of web social media using the data mining technique.
2. Enhancing hate speech classification and finding more accurate results.

3. Implementing a classification analysis for social media dataset that takes from Kaggle and GitHub website, as they are used to solve most problems during the early stages.

1.5 Scope of work

The proposed method will be applied to large-scale data on the Internet which is certified as data set for such topics (hate speech classification), especially on social media sites such as Twitter. The thesis doesn't deal with implicit hate speech, but deal with explicit hate speech.

The proposed system will serve the Iraqi Ministry of Interior as a strong model that can be used in modern approaches for studying the behavior of societies, especially Iraqi society, as a security and scientific basis.

1.6 Contribution of the research

This work plays an important role in classification hate speech in social media using:

a) The Minhash techniques help to re-encoding the data combined with characteristic and signature matrix with TF-IDF in appreciation way;
b) A document-to-vector algorithm with neural network help to classify hate speech with best result;
c) The TF-IDF weight factor which shows a better performance in term of classification compared with stander and other related work;
d) A data mining classifier which is considered a hybrid method with Minhash for hate speech classification.

e) The proposed technique can be used by security authorities such as the Ministry of Interior, as it helps to identify the linguistic trends and verbal behavior that are widespread on Facebook or other social medial platforms.

1.7 Thesis Outline

The thesis is outline into four chapters in addition to Chapter One:

Chapter Two: This chapter presents the data mining techniques and document to vector approach, the approach of Minhash technique was explained, discussing the main quality criterion used for classification within the performance metrics section.

Chapter Three: This chapter presents the proposed system and illustrates the practical stages of the system. It also presents an explanation of the proposed algorithms system within the hate-speech approach.

Chapter Four: This chapter describes the results and evaluates the used system methodology.

Chapter Five: This chapter presents the conclusions that were reached. It also describes possible future work**.**

Chapter Two

Hate speech principles and data mining techniques

2.1 Introduction

Social media provides the possibility of interaction between people despite their different attitudes and directions. The rapid development of websites and social networks provide a variety of tools such as Facebook and Twitter. It contributed to increasing opportunities for extracting and processing large amounts of data, including files, bookmarks, and multimedia like images or videos, that indicate vital communication between users of social media networks. The classification methods varied according to user information in the social network, as it is known that communication networks are heterogeneous and related differently. Given the lack of availability of relationship information in social media, most of the current methods for data processing tend to be unsatisfactory. For this reason, there are many ways to process heterogeneous data, including the social dimension of the inherent affiliations of actors and other tools [34].

Hate speech is one of the important issues of the present time, and it should therefore be taken into consideration by specialists. There are many horrific scenarios that were published on social media, including the Rohingya genocide that occurred in Myanmar, the anti-Muslim mob violence cases in Sri Lanka, and the shootings in Pittsburgh synagogue. So, it has become necessary to develop dynamics to understand how this obnoxious content is spread by social media users [35].

2.2 Social Networks

Social networking sites are a group of websites that rely on internet sites and applications based on personal files for users, where they are allowed to communicate and maintain social relationships around the world by sharing offers and visiting their social accounts. These tools built a lot of personal community sites, discussion forums, and chat rooms through which personal content and networking can be shared, and examples of such sites include Facebook, Twitter, and Instagram [36].

A- Facebook

Facebook is considered the largest and most commonly used platform among social media networks for interaction, such as to answer questions, exchange information, and share daily activities, with more than 1 billion users worldwide. Facebook is a social networking site that makes it easy to connect and share with family and friends online. What makes Facebook unique is the ability to connect and share with the people care about at the same time. For many, having a Facebook account is now an expected part of being online, much like having a personal email address. Due to the increased popularity of Facebook, other websites have been integrated into the platform: a single Facebook account can thus be used to sign in to different services across the Web. Facebook allows users to send messages and post status updates to keep in touch with friends and family. They can also share different types of content, like photos and links. Sharing something on Facebook differs from other types of online communication in that it is often publicly accessible, rather than being private. While Facebook offers privacy tools to help limit who can see the posts that are shared, this social platform was initially designed to be more open and social than traditional communication tools [37].

B- Twitter

Twitter is a free social networking microblogging service that allows registered members to broadcast short posts called tweets. Twitter members can broadcast tweets and follow other users' tweets by using multiple platforms and devices. Tweets and replies to tweets can be sent by cell phone text message, desktop client or by posting at the Twitter.com website. The default settings for Twitter are public. Unlike Facebook or LinkedIn, where members need to approve social connections, anyone can follow anyone on public Twitter. To weave tweets into a conversation thread or connect them to a general topic, members can add hashtags to a keyword in their post. The hashtag, which acts like a meta tag, is expressed as #keyword [38].

Tweets, which may include hyperlinks, are limited to 140 characters, due to the constraints of Twitter's Short Message Service (SMS) delivery system. Because tweets can be delivered to followers in real time, they might seem like instant messages to the novice user. But unlike International Media Support that disappear when the user closes the application, tweets are also posted on the Twitter website. They are permanent, they are searchable and they are public. Anyone can search tweets on Twitter, whether they are a member or not [28].

As is known, social media has become an integral part of our daily life, as well as research and social fields. It is expected that social media tools will play a very significant role in revolutionizing the research methods for building and publishing. It has helped researchers in various fields of communication and linkage to build research, share resources and reach convincing results. The benefits of social media to researchers can be summarized as follows:

- It enabled researchers to communicate with specialists around the world easily
- Researchers can share and collect various information
- They can take benefit from cloud storage and flexible access
- They can promote and support cooperative research methods
- Commercial and business uses

Common social network websites tools for researchers include are in table (2.1) shows the common social tools and their official website [38].

Table 2.1: The common social tools and their official website.

Social Tools	Official Website
Academia	https://www.academia.edu
Res_ID	//researcherid-
Res_Gate	//researchgate-
Insciences	//insciences
Methodspace	methodspace.com
MyScienceWork.com	www.mysciencework.com
Social Science Space	http://www.socialsciencespace.com

2.3 The data types in social network

The data types in social networks are classified into structured, unstructured, and semi-structured data.

2.3.1 **Unstructured data type**: This type of data cannot be identified by its structures, i.e. it cannot be stored in rows and columns in interconnected databases. The data is stored in an unbound way without any defined scheme of storage, for example, storing various pictures and videos from social network sites. One advantage of this type is that it does not require additional components for classifications purposes. However, the limitation of this type is that it cannot control the navigation within the unstructured content. Example for unstructured data are text, audio, images, video, web chats, logs, and so on [39]. The datasets in our work on Twitter is a type of

Unstructured data. The Figure (2.1) shows Unstructured Database types [40].

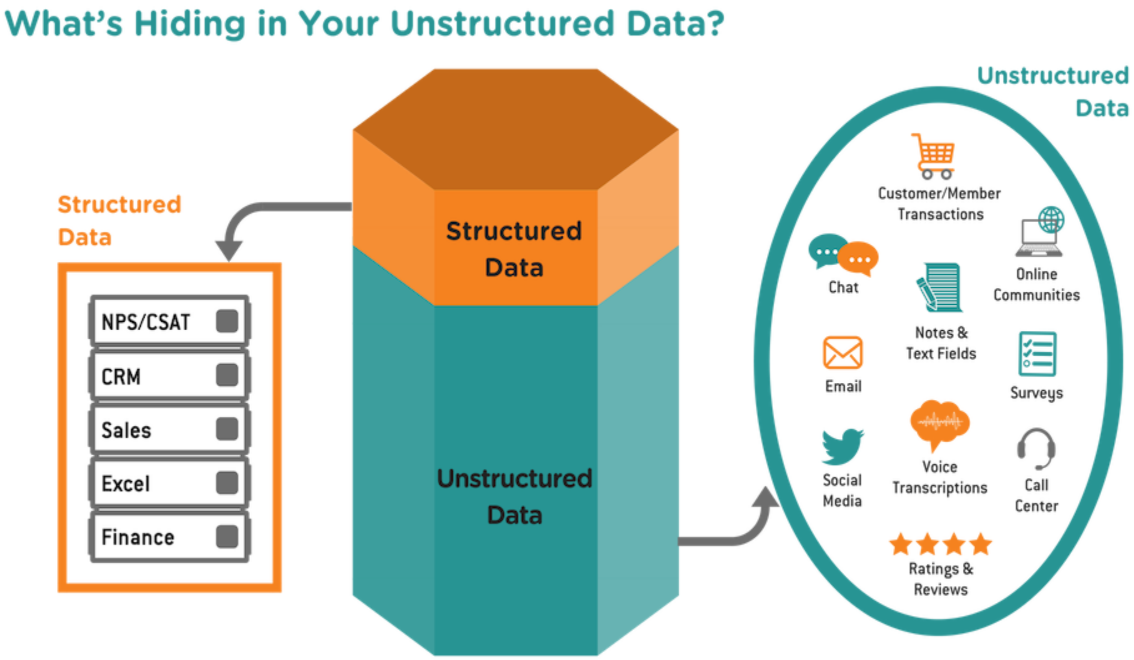


Figure (2.1): Unstructured Database [40].

2.3.2 **Structured Data type**: This type of data in social media networks is based on a graph structure as a simple framework. The data is described in the scheme of a graph G = (V, E), as the V is a collection of nodes/objects, for instance, people, organizations, or products, while E represents a group of edges or relationships for nodes connect response patterns. Measuring and dealing with this type of data is done by analyzing the social network. As well, applications that are based on graph analysis based on the intelligence methods of correlated data [41]. The Figure (2.2) illustrates the concept of relational structured data base type.

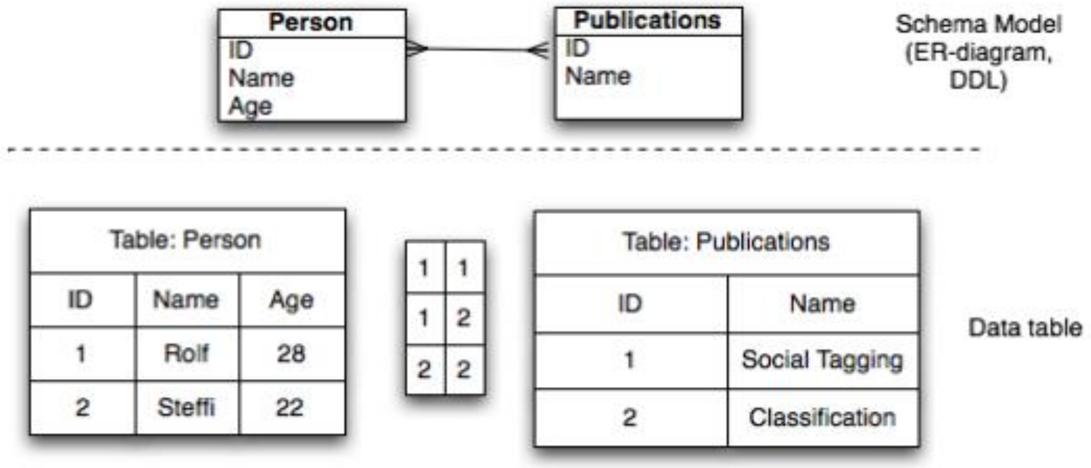


Table: Person		
ID	Name	Age
1	Rolf	28
2	Steffi	22

1	1
1	2
2	2

Table: Publications	
ID	Name
1	Social Tagging
2	Classification

Figure (2.2): Structured Relational Database [41].

2.3.3 Semi-structured data: It is considered a special case of structured data that does not depend on a formal and traditional structure to represent data models for interconnected data, and therefore contains signs and other evidence for the separation of semantic elements as well as the adoption of a hierarchy to represent data. For example, the data obtained through e-mail and deposit data is often represented in specific forms such as JavaScript Object Notation and Extensible Markup Language [42]. The Figure (2.3) shows the Resource Description Framework graph representation as a form of semi-structured data type**.**

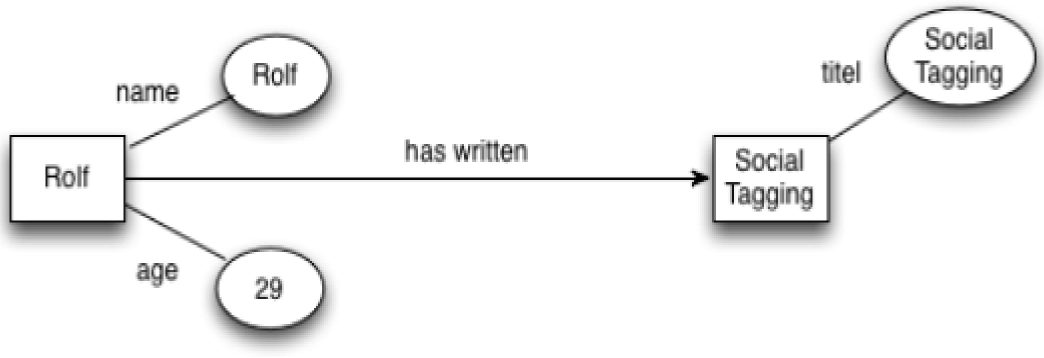


Figure 2.3: The RDF Graph Representation [42].

2.4 characterization of the dataset

Several tests were conducted on a set of data to distinguish hate speech and to understand the accuracy of the proposed strategy, and are explained as follows:

1- The first dataset was declared in "github4" So, the tweets that are covered in the used are categorized into one of the following three sections, presented as the "Sexism/Racism/Neither". The first set considered as "Sexism/Racism" are related to particular "hate-speech" classes, and are therefore combined as a hateful class portion [43].

2- The second dataset was shared by "Crowd-Flower" and included "24,783" tweets of the English language categorized into three classes: "Hate" which involves tweets that include hate speech, "Offensive", which involve tweets including attacking language without hate speech, and the third one is tweets of "Neither" types. The size of the dataset is somewhat small, but it will still be applied for this research [43][44].

3- The third data set consists of 31,935 tweets in csv format and has been obtained from the global site Kaggle. The data is divided into 93 percent (29,695 tweets) which present Twitter details of the non-hate class, and 7 percent (2,240 tweets) that reflect Twitter details of the hate class [45].

4- The fourth dataset was declared on the Kaggle website and had been employed within the work of [46]. It consists of 1,600,000 tweet samples obtained through the Twitter-API.

2.5 Data Mining Techniques in Social Networks

Data mining Provides many technologies to discover the knowledge of the vast amount of big data on social media. The data mining techniques are used for data retrieval and statistical modeling as well as for purposes of machine learning. These technologies provide multiple capabilities for data analysis purposes pre-processing data, interpretation processes, and analyzing data. Data mining techniques are capable of managing the three principal disputes by social network data as the size, the noise, and dynamism. The nature of social network voluminous datasets requires automated information processing for examining social data within a reasonable time [47].

There are many steps that data mining uses to reach the ultimate knowledge, these steps are as follows:

1- Data selection

It is considered one of the important processes in retrieving related data related to the task of data analysis by retrieving the data from the database. Most of the time, data conversion and merging are performed before the data selected from the databases.

2- Data Preprocessing

It is an important step in the data mining process as it requires finding solutions to data collected from more than one source and is often incompatible from multiple environments and within different periods where several methods are used to integrate heterogeneous data in order to identify events and combine them within the data warehouse, and data preprocessing implements with another methods as data cleaning(the process of identifying inaccurate and inconsistent data to improve data quality and clean up data from errors), data integration (It is represented in the process of providing a unified presentation of data obtained from more than one data source in a database and flat files as well as data in multiple data cubes), data reduction (this techniques used for the purposes of reducing the vast amount of data used for exploration purposes by a low representation of this data as this data is small in size but still contains important information), while data transformation as it is the process of converting data from one format to another format and it is considered one of the important processes for data integration and management purposes as it allows data integration and the integration of data applications [48].The figure(2.5) shows the preprocessing steps.

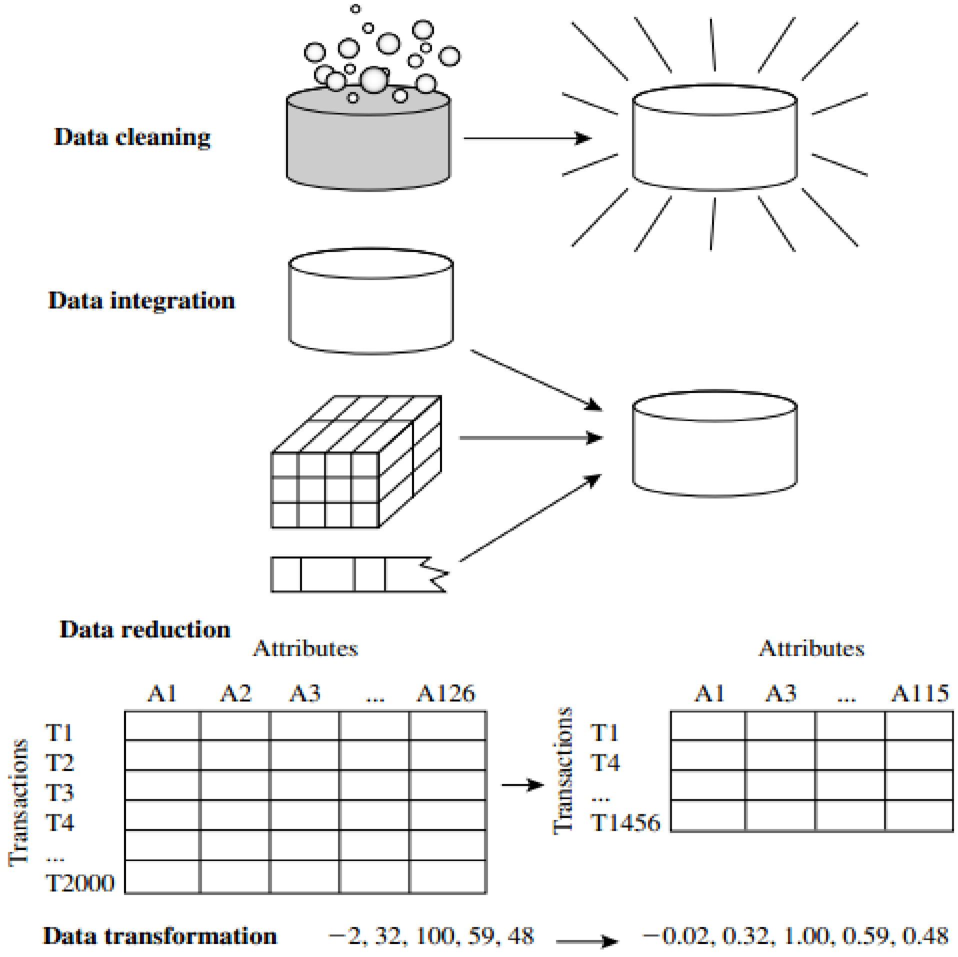


Figure (2.4): data preprocessing methods [48].

3- Pattern Identification

The process relies on data exploration and definition of different variables, where specific patterns are defined that make the prediction of knowledge optimal [49].

4- Interpretation Evaluation

It is considered one of the most necessary stages because the data after completing its processing, examination, and production requires examining the outputs for the purposes of interpreting and evaluating the different patterns, and thus it is possible to find new views that provide solutions to the problems that have been analyzed. While the last step explained as the knowledge presentation state to represent the data. The data is in multi-dimensional space to obtain knowledge based on all the previous steps [50]. The figure (2.6) shows the knowledge discovery state within data mining.

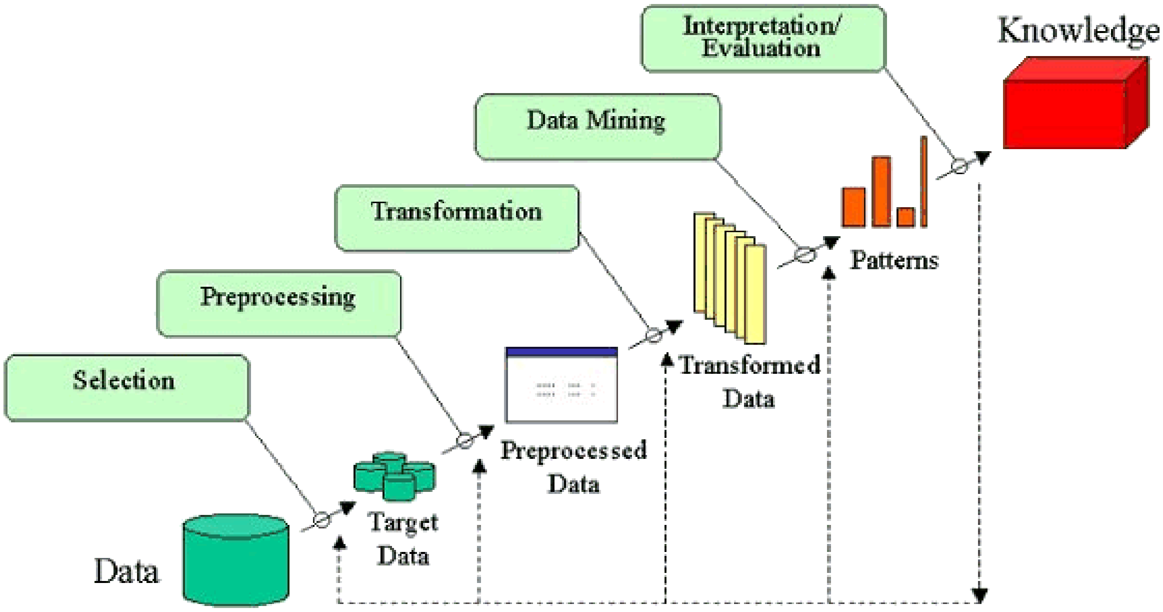
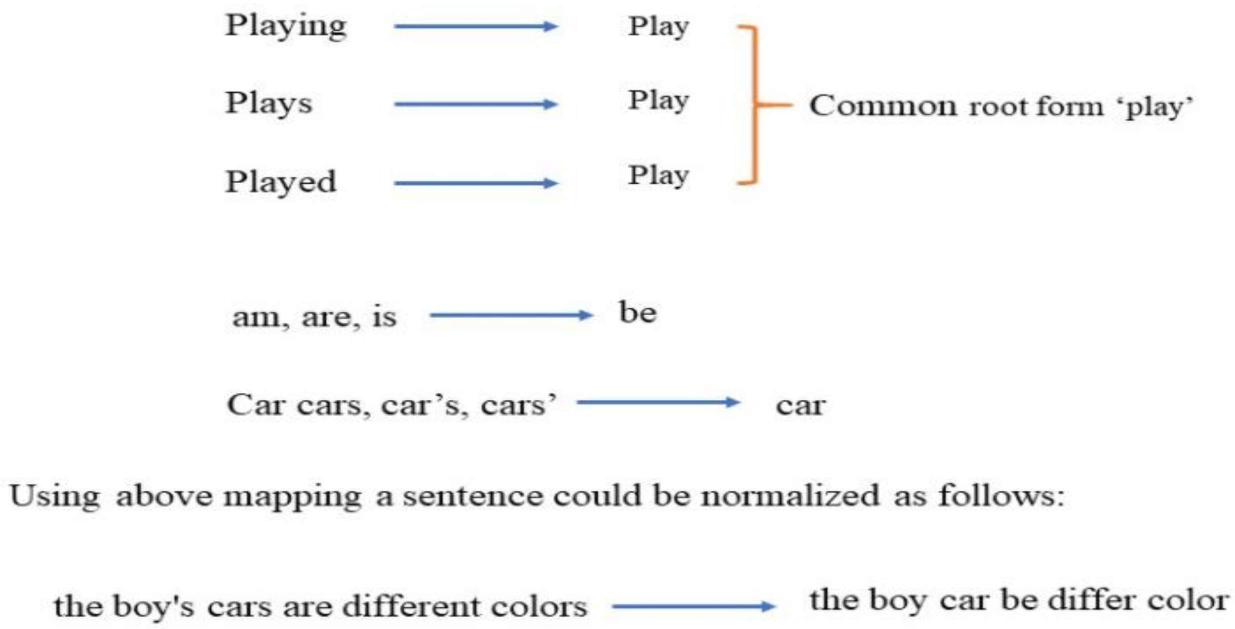


Figure (2.5): Knowledge discovery state of data mining [50].

2.6 Word Normalization

Stemming and Lemmatization are text normalization techniques within the field of Natural language Processing that are used to prepare text, words, and documents for further processing. In this section, you may study stemming and lemmatization in an exceedingly practical approach covering the background, applications of stemming and lemmatization, and the way to stem and lemmatize words, sentences and documents using the Python Natural Language Toolkit package which is the natural language package provided by Python In natural language processing, You will want to remember your software, that the words “kick” and “kicked” are just different tenses of the same verb. this can be the concept of reducing different kinds of a word to a core root [51]. Figure (2.6) shows the main operation of word Stemming



Using above mapping a sentence could be normalized as follows:

Figure (2.6) Word Normalization [51]

2.7 N-gram and Minhash Techniques.

these techniques are very important models in dealing with text analysis based on mathematical operations, and the concepts have been clarified in the following sub-sections:

2.7.1 The N-gram

The N-gram concept was introduced for the first time by Shannon (1948) on the topic of information theory in general, as he wanted to expand the concept of general theory used to communicate through. It was explained that the messages are related to each other to complete the meaning according to specific physical and conceptual entities and that the actual message is derived from a group of possible messages. The system must therefore be designed in such a way to be able to adapt and work for each possible choice. Shannon used N-grams for analyzing and predicting printed English [52].

The K-shingle or n-gram that spilts the text into a set of substrings depends on the value of the selected K of length [53], and the numbers of shingling (sum of a substring in all tweets) determined by equation(2.1)

$$\left((n-k)+1\right) \ldots (2.1)[53]$$

where n is the number of words in the tweet, and k is the length of the shingles number. For example, taking the tweet *"These women don't care they just play that role"* and supposing that k=4, in this case the k-shingling will be (These women don't care, women don't care they, don't care they just, care they just play, they just play that, just play that role). This example indicates that the number of shingles is 6, i.e. by applying ((9-4) +1).

2.7.2 Minhash Technique

This method has been discovered by Andrei Broder. It was initially used for AltaVista search engine purposes to detect the presence of duplicate pages on the web and remove them from search results, as well as in the process of solving the problems of grouping on large-scale documents by finding similarities between the vocabulary group for documents [54].

MinHash is a datamining algorithm used to find similarity estimations between tweets. The main property to the algorithm is that it has many hash functions. Some concepts of the Minhash techniques are applied in this thesis. The general form of the Minhash is presented in Equation (2.1) [55].

$$h(x)= (a\,x+b) \bmod c \quad \ldots\ldots. \quad (2.1)\ [55]$$

Where **(a** & **b)** are two numbers whose values are random, **x** is hash value, and **c** is a prime number that is larger than the maximum number of the total single set. In this work, the MinHash technique uses many hash functions to generate hash keys that refer to each class within the dataset that has been used.

2.8 The MD5, crc32 and SHA hashing techniques and their main characteristics.

A- Message Digest 5 (MD5)

The MD5 hash functions usually use a hexadecimal number equal to 32 digits, which is represented as an n-gram of two documents, for instance T1, T2, Sn(T1)-Sn(T2). The Jaccard of coefficient, as shown in Equation (2.2) below, can be applied to calculate the similarity of the two-documents [56].

$$\mathrm{sim}(T_1, T_2) - \frac{|S_n(T_1) \cap S_n(T_2)|}{|S_n(T_1) \cup S_n(T_2)|} \ldots (2.2)[56]$$

This function relies on a threshold to determine the duplicates between the two tweets T1 & T2. Besides, the size window (n) and the value of the threshold is determined based on the experiments.

One of the most important characteristics of the MD5 algorithm is the ability to create hash value from an input value. It provides an integrity check on the data that is transmitted between servers and common characteristics of MDF are described as four rounds. It has a unique additive constant; besides that, it creates a 128-bits message from the input data as 32 digits hexadecimal number. Figure 2.7 presents an example for the MD5 representation of input data [56].

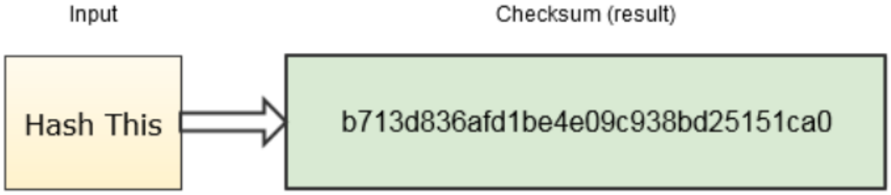


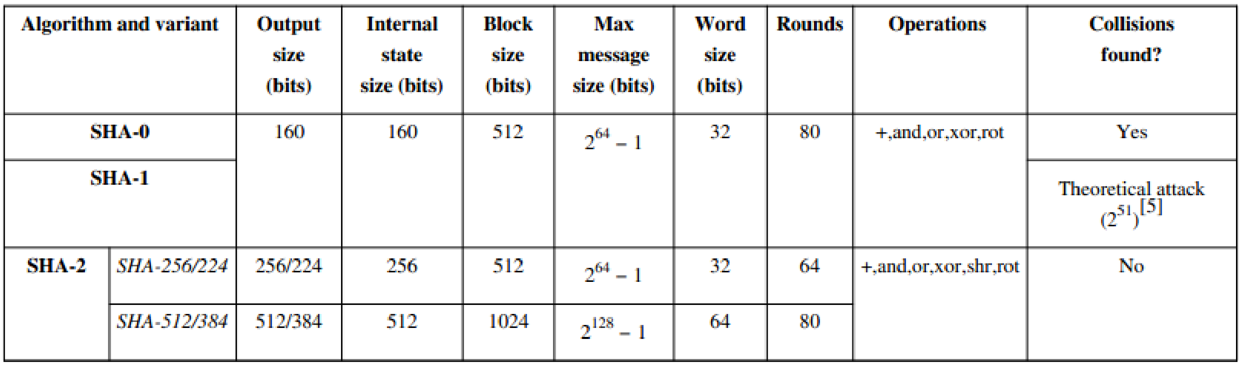
Figure 2.7: MD5 input data with Hexadecimal Representation [56].

B- Secure Hash Algorithm

It consists of four main standard algorithms as (SHA_0,1,2,3) that have different structures, provided by the National-Security Agency (NSA) and published by the NIST.

SHA_1 is considered to be the most common among the rest of the types, as its work relies on messaging purposes of 160-bit message digest, and the highest value of the length represented 264. The algorithm work depends on the message summary, which is much smaller compared to the message itself and thus provides a significant improvement in performance. The SHA_0 is used for a short period due to a major defect that was not disclosed, and was later modified with the version it represented by SHA_1. Furthermore, SHA.3 holds the same length as SHA_2, and the structure of the algorithm internally differs significantly from the other SHA family. The SHA_2 family operates by means of a similar algorithm by an irregular digest size [57]

Table 2.2: A comparison among SHA-x algorithms based on common comparison parameters [57].



Algorithm and variant		Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Word size (bits)	Rounds	Operations	Collisions found?
SHA-0		160	160	512	$2^{64} - 1$	32	80	+,and,or,xor,rot	Yes
SHA-1									Theoretical attack (2^{51})[5]
SHA-2	*SHA-256/224*	256/224	256	512	$2^{64} - 1$	32	64	+,and,or,xor,shr,rot	No
	SHA-512/384	512/384	512	1024	$2^{128} - 1$	64	80		

One of the characteristics of the SHA_x group of algorithms is their ability to secure data by providing encryption with different properties, for instance, pre-image, and collision resistance.

This makes it difficult for the attacker to know the original content during the long period, caused by the encryption of this algorithm. Another characteristic is the difficulty to obtain two various messages that contain hash to the exact hash-value [58].

C- cyclic redundancy check CRC32

CRC is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data. Blocks of data entering these systems get a short check value attached, based on the remainder of a polynomial division of their contents. On retrieval, the calculation is repeated and, in the event the check values do not match, corrective action can be taken against data corruption. CRCs can be used for error correction [59].

CRCs are so called because the check (data verification) value is a redundancy (it expands the message without adding information) and the algorithm is based on cyclic codes. CRCs are popular because they are simple to implement in binary hardware, easy to analyze mathematically, and particularly good at detecting common errors caused by noise in transmission channels. Because the check value has a fixed length, the function that generates it is occasionally used as a hash function.

2.9 TF-IDF Weight Factor

Two key values were collected by the TF-IDF weight.the uniform phrase frequency (TF), corresponds to the times a word appears in a text, as divided by the total number of terms in the document. Secondly, the Inverse Document Frequency (IDF) is measured as the document number logarithm divided by the number of times a given word (T) appears in the dataset. In general, TF-IDF uses the value of a term in a document depending on the time this term appears in that document, in addition to the collection of documents granted. If a word takes place regularly in a text, the intuition for this measure means that it is significant and specific, and would assign a higher score to that word. However, since this word often appears in other records, it is definitely not a special word, then a lower score can be given [60].

- **TF: Term Frequency**: it computes the frequency of a term shown in a particular document. The text in the dataset is different in size and duration. Since a word will occur more frequently than the shortest duration in long documents, thus the term frequency break by the duration of the text. TF is estimated using Equation (2.3) below.

$$\boldsymbol{Tf} = \frac{\boldsymbol{Term's\ count\ of\ times\ apperars\ in\ a\ document}}{\boldsymbol{Terms\ total\ count\ in\ a\ document}} \text{ ...(2.3) [60]}$$

- **IDF: Inverse Document Frequency**: It measures how important a term is in a document. Through TF computing process, all terms are equally important. Yet, in some cases there are some known terms that occur much frequently in the document but have little importance. Therefore, a mechanism is needed to weigh down the repeated terms while scaling up the uncommon terms, by computing the IDF as shown in Equation (2.4):

$$\boldsymbol{IDF(t)} = \frac{\textbf{log document's total number}}{\boldsymbol{Number\ of\ documents\ has\ term\ (t)}} \ldots .\ (2.4)\ [60]$$

Thus, to clarify, for a term t in a tweet d, the W_t indicates weightiness, and d of representation t in tweets d is presented by:

$$W_{t,d}=TF_{t,d} \log (N/DF_t) \text{ ------- } (2.5)\ [60]$$

Where:

- $TF_{t,\ d}$ is the number of occurrences of term t in tweet d.
- DF_t is the number of tweets holding the term t.
- N the total number of tweets within the frame.

For example, if we have three tweets T1, T2, T3 the content of the tweets is

T1 = (The sky is raining profusely)

T2 = (The sky is raining profusely and snow)

T3= (the sky is thunder strongly), table (2.3) shows example of TF-IDF.

Table 2.3: TF-IDF Example with k=3

Shingles	T1	T2	T3
The sky is	**1**	**1**	**1**
Sky is raining	**1**	**1**	**1**

Is raining profusely	**1**	**1**	**0**
raining profusely and	**0**	**1**	**0**
profusely and snow	**0**	**1**	**0**
sky is thunder	**0**	**0**	**1**
is thunder strongly	**0**	**0**	**1**

After implementing each of (Eq.2.3 and Eq.2.4) on the previous state, we have got the following result in table (2.4):

Table 2.4: TF-IDF Example results with k=3

Shingles	T1	T2	T3
The sky is	**1.5**	**0.4**	**0.4**
Sky is raining	**0.4**	**0.17**	**0.17**
Is raining profusely	**0.4**	**0**	**0**
raining profusely and	**1**	**0.4**	**0**
profusely and snow	**0**	**0.4**	**0**
sky is thunder	**0**	**0**	**0.4**
is thunder strongly	**0**	**0**	**0.4**

2.10 Document to vector (DOC2VEC)

The Embeds are generated using the Doc2Vec model. The goal is to have a representation of a tweet vector. Any simple pre-processing of data is added before implementing Doc2Vec, such as deleting stop words and punctuation, as well as translating all text to lowercase.

Doc2Vec is a template of the current models known as Word2Vec that was created in 2014, producing vector word representations. By merging individual word vectors, Word2Vec represents documents, but by doing so it loses all word order information, In Word2Vec, Doc2Vec is extended by applying to the output representation a "text vector" that includes some information about the document as a whole and enables the type to define some word order information. Maintaining order details makes Doc2Vec useful for exp loring the tiny changes between text documents in the proposed model [61]. There is tow type of document to vector and Bag of Words version of Paragraph Vector and Distributed Memory version of Paragraph Vector.

First should learn how to the concept of distributed vector representation of words in figure (2.8) The purpose is to predict a word in a context provided the other terms [61].

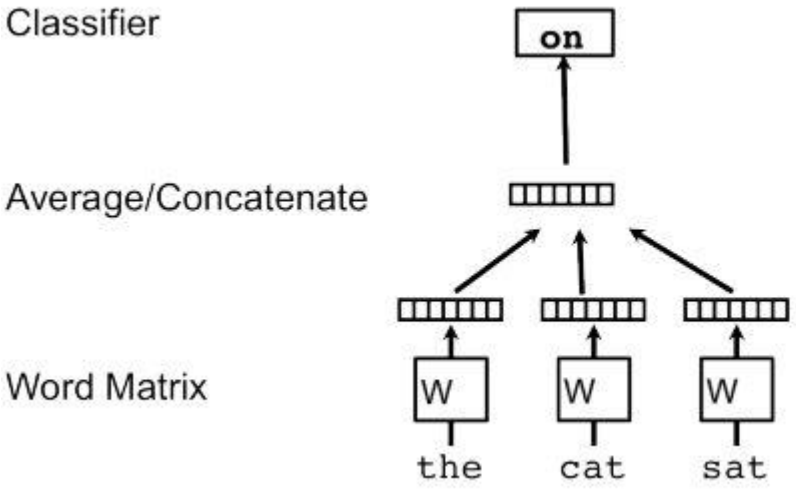


Figure (2.8) word to vector

Each term is mapped to a unique vector, represented in a matrix W by a column. The column is indexed according to the word's place in the vocabulary. The concatenation or sum of the vectors is then used as characteristics in a sentence to predict the next term. More formally, the target of the word vector model is to optimize the average log likelihood, given a sequence of training terms w_1; w_2; w_3,……, w_T in equation (2.6). Typically, the prediction task is performed using a multiclass classifier, such as SoftMax equation (2.7). There we have for each output term i each y_i is un-normalized log-probability, computed as equation (2.8) Where the parameters of SoftMax are U; b. A concatenation or average of word vectors derived from W is built by h.

$$\frac{1}{T}\sum_{t=k}^{T-k} \log p(w_t \mid w_{t-k}, \dots, w_{t+k}) ----(2.6)$$

$$p(w_t \mid w_{t-k}, \dots, w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} ----(2.7)$$

$$y = b + Uh(w_{t-k}, \dots, w_{t+k}; W) ----(2.8)$$

figure (2.9) Each paragraph is mapped to a unique vector, represented in matrix D by a column, and each term is mapped to a unique vector, represented in matrix W by a column. To predict the next term in a sense, the paragraph vector and word vectors are averaged or concatenated. Document to vector use concatenation as the strategy to merge the vectors in the experiments. The only difference will be made about a topic word to vector in equation (2.8) Where h is generated from W and D.

There is two type of Doc2vec [61]

A- (PV-DM)

All sentence vectors and word vectors are initialized randomly in this form. Each paragraph vector is allocated to a single text, while all documents share word vectors. By way of back propagation, either averaging or concatenating all paragraph vector and word vectors and passing to stochastic gradient descent and gradient is obtained. This method is similar to the word2vec continuous bag-of-words approach [61] figure (2.9) shows PV-DM.

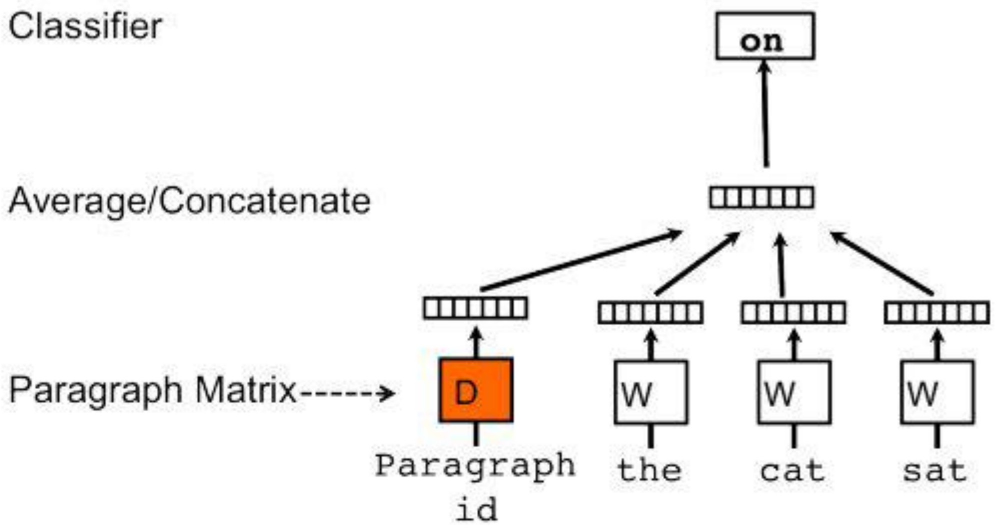


Figure (2.9) Shows PV-DM

The concatenation or average of this vector with a three-word background is used in this model to predict the fourth word. The paragraph vector represents information that is absent from the current context and can serve as a memory of the paragraph subject.

B- (PV-DBOW)

It is defined as the way to disregard the input background terms, but forces the model to predict randomly sampled words from the output paragraph. What this basically means is that system sample a text window at each iteration of Stochastic gradient descent, then sample a random word from the text window and form a classification task given the Paragraph Vector. Figure (2.10) illustrates this strategy.

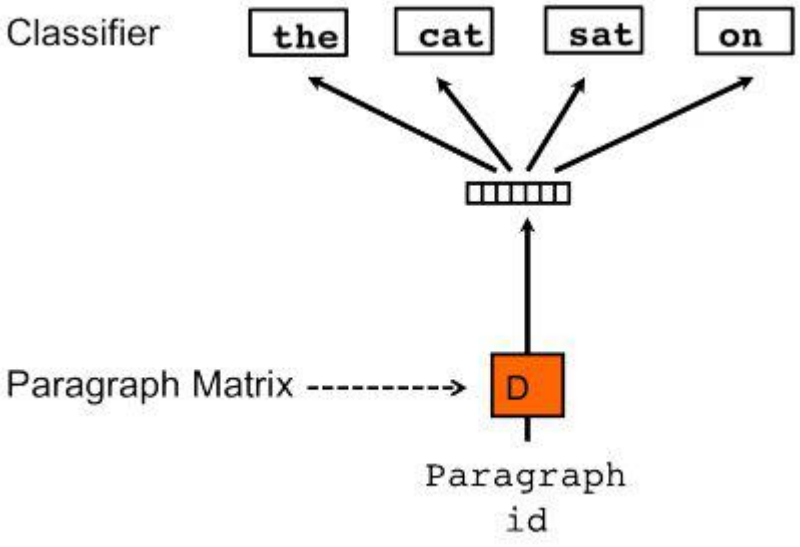


Figure (2.10) shows PV-DBOW

The paragraph vector is learned in this form to predict the words in a small window. In addition to being conceptually clear, fewer data must be processed in this model. As compared to both SoftMax weights and word vectors in the previous model, only can store the SoftMax weights. This model is also equivalent in term vectors to the Skip-gram model [62]

2.11 Data mining classifiers

Data mining provides many technologies to discover the knowledge from the vast amount of big data on social media. Data mining techniques are used for data retrieval and statistical modeling for purposes of Machine Learning (ML). These technologies provide multiple capabilities for data analysis purposes including pre-processing data, interpretation processes, and analyzing data. Such techniques manage the principal disputes by social network data as the size, the noise, and dynamism principals. The voluminous nature of social network data-sets require information processing that is automated for the examination of social data [63].

There are many models that are used for machine learning to assess performance, and the most common of them have been adopted as follows:

A- Logistic Regression (LR): Logistic regression is a form of regression used to predict the likelihood of a binary output from an input dataset modeled using (2.9) [64].

$$P = \frac{1}{1 + e^{-z}} \dots (2.9)$$

where P is the frequency probability of the case and can take values between 0 and 1 and produce an S-shaped curve z the linear combination of the variables observed corresponds to that set in X. The Linear Regression model equation is defined by (2.10).

$$z = \theta_0 + \theta_1 x_1 + \cdots + \theta_i x_i \dots (2.10)$$

where θ_0 is the intersection point of the model θ_i belongs to the regression coefficients and x_i to the independent variables for the logistic regression model.
Thus, the logistic regression model allows to determinate an event occurrence, the regression model can determinate if the autoclave is into a specific stage of the sterilization process.

B. Support Vector Machine (SVM): It is a linear algorithm for the problems of regression and classification. The idea of SVM is simple: The algorithm creates a line or a hyperplane that separates the data into categories [64]. One of the advantages of this algorithm is the efficient handling of memory, but it does not work well with big data [65]. SVM is regarded as the classifier that maximizes the difference between certain important vectors or support vectors. This method is more likely to result in a stronger data separation because the foundation lies in optimizing the distance between the support vectors that provide the ideal hyperplane. The basic Support Vector Machine uses a linear decision threshold

Data is linearly separated. However, it is attempting to reach a threshold that increases the margin (M) requiring the use of an optimizing procedure, with a restriction that all points be on the correct part of the hyperplane [66].

$$y = w^T x_i + b = 0 \quad (2.11)$$

$$y = w^T x_i + b = 1 \quad (2.12)$$

$$y = w^T x_i + b = -1 \quad (2.13)$$

Can extract the margin M as follows:

$$M = \frac{(1-b)}{||w||} - \frac{(-1-b)}{||w||} = \frac{2}{||w||} \qquad (2.14)$$

Here

- M : the margin
- y : predicted class label ∊ –[1, 1]
- x_i : i^{th} attribute
- w_i : weights
- b : bias

C. Random Forest (RF): Random forests are an outfit learning strategy by developing a large number of decision trees at preparing time and produce the class. It is ideal for decision trees to overfit their training set. The random forest simulation algorithm applies the common bootstrap strategy of aggregating to tree learners. Instead of a training set $X = x1, \ldots xn$ with $Y = y1, \ldots yn$, bagging frequently (B times) picks a random sample to substitute the training set and matches trees for $b = 1, \ldots, B$ to such samples. Examples, with substitution, n illustration of X, Y training; name these X_b, Y_b. Train a tree f_b on X_b, Y_b. for classification or regression. Predictions for unknown samples x'can be produced after training.

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x') \qquad (2.15)$$

Where, $\hat{f}$ is predictions from each tree, B is bagging repeatedly, b is sampling for $(b = 1, \ldots, B)$, f_b is regression tree and x' is unseen samples [67].

D. Naive Bayes (NB): This is a Bayes Theorem based method in mathematical classification. This is one of the easiest algorithms for supervised learning. Classifier Naive Bayes is a quick, accurate, and reliable algorithm. Naive Bayes classifiers on broad datasets provide high accuracy and speed [68]. It is one of the powerful and scalable learning algorithms for machine learning and data mining, with strong predictive efficiency. It can be trained successfully in a supervised environment. This often includes minimal training data, and may also be used with various attributes for small and broad training data set. This classifier can accommodate incomplete values. Classifier claims that the attributes relate individually to the likelihood of the model that is described in the formula that follows:

$$P(c|F) = \frac{P(F|c)\,P(c)}{P(F)} \qquad (2.16)$$

Where P is a probability, variable c is the dependent class label and F are the several feature variables [68].

E. Feed-forward Neural Network

neural networks are strong learning models. Feed-forward networks involve networks with fully connected layers, such as the multi-layer hidden layer, as well as networks with convolution and pooling layers. The computing system of the brain, which consists of computational units called neurons, is inspired by neural networks. A neuron in the metaphor is a computational device of scalar inputs and outputs. There's a related weight for each input. Each input is multiplied by the neuron by its weight, then summed, added to the outcome by a non-linear function, and transferred to its output. The neurons are connected to each other, creating a network: a neuron's output will feed into one or more neurons' inputs. These networks have been found to be highly efficient computing machines. A neural network with sufficiently neurons and a nonlinear activation function will estimate a very broad variety of mathematical functions if the weights are set correctly [70] figure (2.10) shows the Architecture of neural network.

$$h_1 = \text{ReLU}(W_1 x + b_1) ----(2.17)$$
$$h_2 = \text{ReLU}(W_2 h_1 + b_2) ----(2.18)$$
$$y = \text{Logits}(W_3 h_2 + b_3) ----(2.19)$$

- x is the input
- y is the output (what we want to predict)
- h is the hidden layer
- W are the parameters of the hidden layer

2.12 Performance Metrics

Within the used system, the performance of each method has been evaluated by employing precision and macro averaged F1 score. There is no particular arrangement within writing about which the evaluation measurements apply. In any case, it is considered that collecting both precision and large scale F1 offers great bits of knowledge into the relevant qualities and shortcomings of each method.

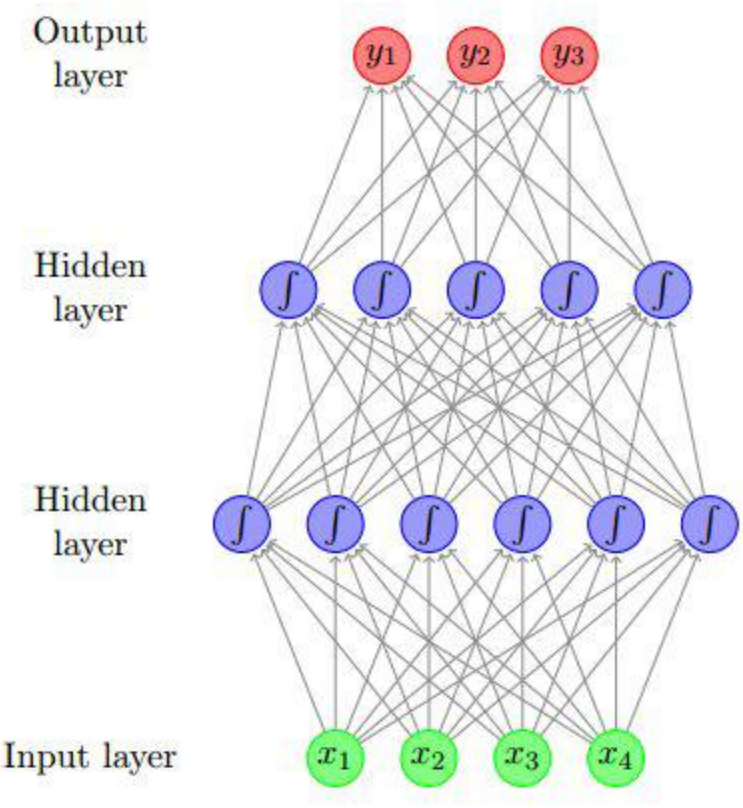


Figure (2.11) Architecture of neural network

Table (2.5) Confusion matrix for binary classification .

Actual	**Predicated**	
	Normal	**Hate**
Normal	**TP**	**FN**
Hate	**FP**	**TN**

Based on the values in Equation (2.20), two measurements were created. One known and perhaps most straightforward metric is the accuracy that measures total correct expectations as a percentage of total cases [71]. It is defined as follows:

$$\text{Accuracy} = \frac{tp+tn}{tp+fn+fp+tn} \qquad \dots (2.20)$$

However, looking only at accuracy to determine if a model is good is not sufficient enough to draw such a conclusion. Another well-known scale of classification problem is the F1 score, which is represented as the harmonic-mean of accuracy and summons [71]. It is shown as:

$$\text{Recall} = \frac{tp}{tp+fn}$$

$$\text{Precision} = \frac{tp}{tp+fp}$$

$$\text{F1} = 2 * \frac{recall * precision}{recall + precision}$$

In case of an unbalanced dataset, the F1 score will give an unbiased performance representation that is more suitable than the accuracy measurement.

2.13 Summary

This chapter presented the general social network tools for researching and society sides explained, in addition to the data types in social network and their characteristics. Descriptions of both N-gram and Minhash techniques have been presented. Besides, the hashing techniques as MD5, SHA and CRC32 and their main characteristics have been explained. In addition, the concept of D2V was discussed and what are its modules. the different data mining techniques used in social networks were also taken into consideration.

Chapter Three

THE PROPOSED METHODOLOGY

3.1 Overview

Chapter three includes the main concepts of this thesis. It explains in detail the techniques behind hate speech detection system in order to show which technique is the best to use in improving future detection systems for hate speech and violence languages. The outlines of this chapter can be summarized in the following way: the first section describes the data used as well as how it was entered into the system and converted it into an acceptable form. The second section illustrates the architecture of cleaning data and stemming system in detail, with all the related algorithms and theories. The third section proposes the utilized types of data (Twitter dataset), k-shingle technique, a hash of tokens, TF-IDF, Minhash, document to vector (D2V) technique and data mining classifiers (RF-SVM-LR- Naive Bayes -neural network) with evaluation measurement. Particularly in this section of the thesis, five data mining algorithms will be discussed, as these are used for the classification of data.

3.2 Proposed Methodology in general

The structure of the proposed system consists of a number of stages that all cooperate to propose a technique for detecting hate and violence speech in social media speech. The flow chart in Figure (3.1) illustrates the system architecture and describes the steps involved in the hate detection system that differ from traditional methods. The data used in this system is ready data of various sizes from reliable sites on the Internet.

The second stage of the proposed system is the data cleaning process which removes all white-spaces and punctuation marks, and changes the capital letters into small ones, in addition to removing all tags and data that are unnecessary in order to simplify the process throughout the following steps.

The third stage involves three algorithms to deal with the data. First, the Minhash technique is responsible for generating a new matrix with new values, in form of a number of hashes and tweets. The number of hashes can take from 1-20 hash, the number of classes of original data set has been taken, which is the same number of hashes to Achieve high accuracy, while the number of tweets reflects the actual number of tweets, on which the similarity to be verified is based. Secondly, converting words to vectors, or word vectorization, is a natural language processing (NLP) process. The process uses language models or techniques to map words into vector space, that is, to represent each word by a vector of real numbers. Meanwhile, it allows words with similar meanings to have similar representations. The third algorithm is the TF-IDF. Therefore, a certain way is needed to weight the effects of these frequent terms, and to weight up the effects of the terms that are less frequent in the documents. The use of logarithms has solved this problem, as will be explained later.

The fourth stage deals with the data mining classification. The classifier operation is a term utilized both recognizing and circulation sorts of "things" into various gatherings and for the subsequent arrangement of classes, just as the task of components is to pre-setup classes. To characterize the wide importance given above is an essential idea and a piece of practically a wide range of exercises. Five classifier procedures have been used, namely Naive Bayes algorithm, Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), and Neural network. In the proposed solution, the classifiers that achieves high result have been chosen which are RF and neural network.

Four criteria have been used throughout the five-stage continuous device assessment in order to test the Accuracy, Precision, Recall & F1 Score device: Success Measure Interpretation. Accuracy is the most intuitive output indicator which is essentially the ratio of the cumulative input to a correctly discovered note. Precision is the ratio of positive feedback that has been predicted accurately to the overall predicted positive feedback. Recall (sensitivity)-recall is the percentage of accurately expected positive input to all real class results-indeed, and the F1 score is the precision and recall weighted average.

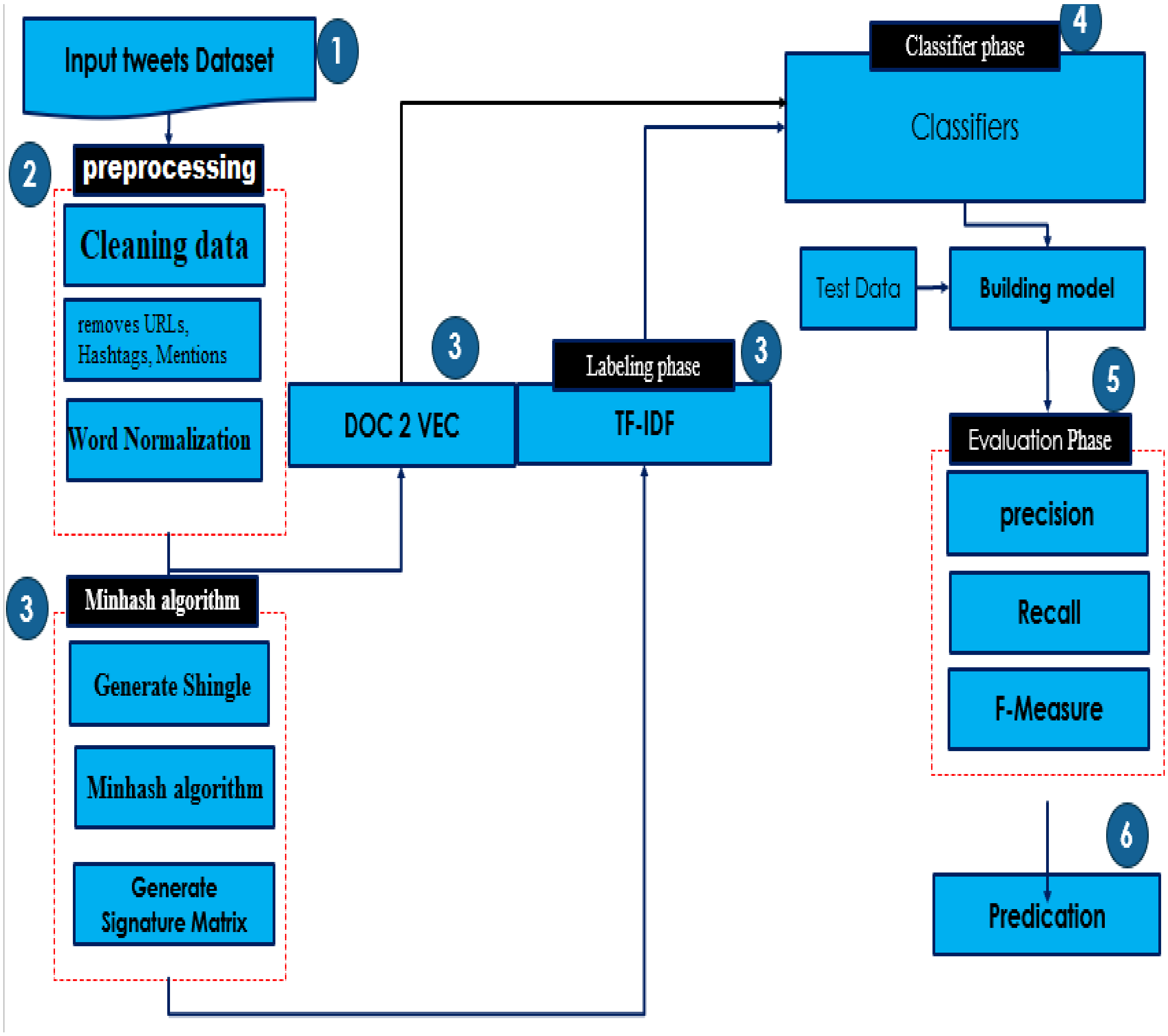


Figure (3.1) System general diagram

3.3 Dataset description

Several tests were conducted on a set of data to distinguish hate speech and to understand the accuracy of the proposed strategy, and are explained as follows:

1- The first dataset that mentioned it in chapter (2.4.1). Figure (3.2) shows the distribution of tweets to each classes whereas the tweets of the class "Neither" have been rejected due to its lack of implication. Various tweets of this type were obtained and manually verified, and they have been recognized as relating to both classes.

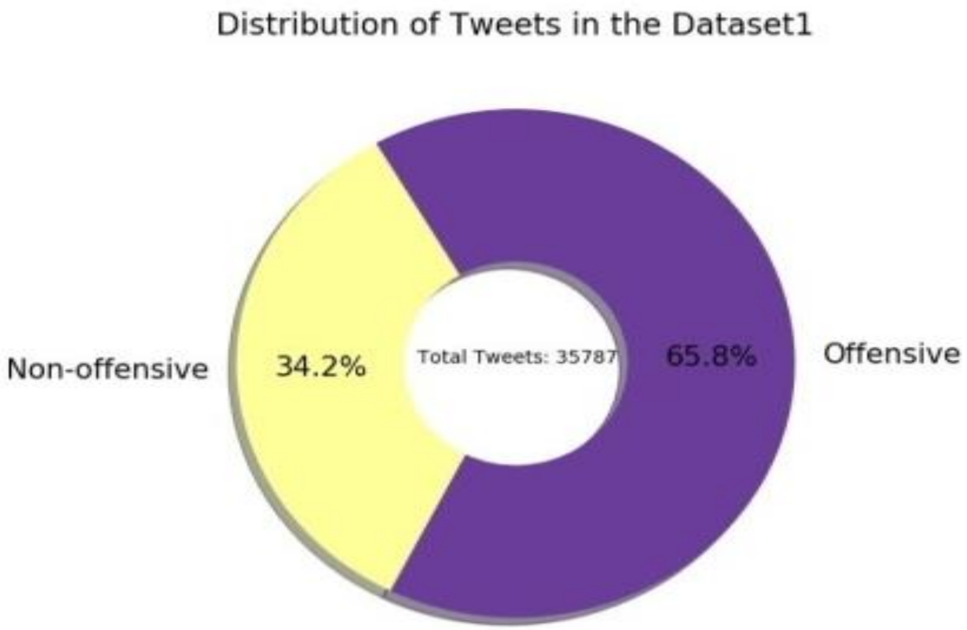


Figure 3.2 Description of Dataset 1

2- The second dataset that mentioned it in (2.4.2). The combination of the three tweet categories is explained in Figure (3.2). The numbers indicate that about 5% of the tweets include hate speech, while the majority tweets (77%) include attacking language. This means that the number of tweets relating to the three classes is actually skewed, leading to an unreliable dataset.

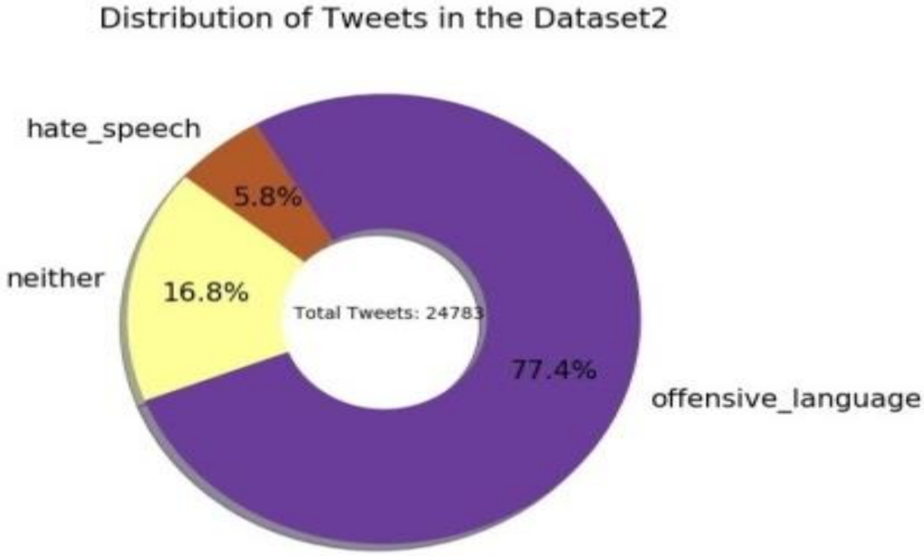


Figure (3.3) Description of Dataset 2

3- The third data set consists of 31,935 tweets in csv format that mentioned (2.4.3). Figure (3.4) shows the distribution of tweets according to each class in addition to the number of tweets in each class.

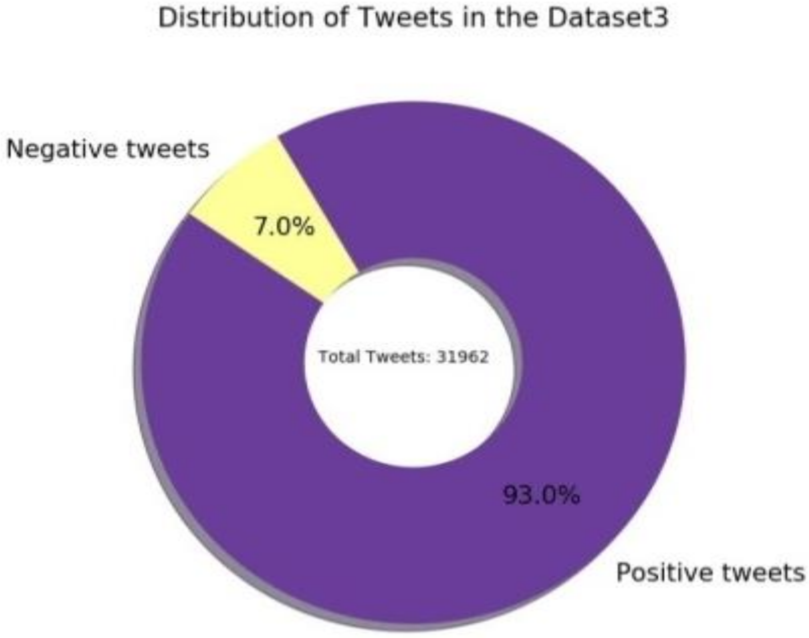


Figure 3.4 Description of dataset 3

4- The fourth dataset that mentioned it in (2.4.4). Figure (3.5) shows the data divisions according to each class.

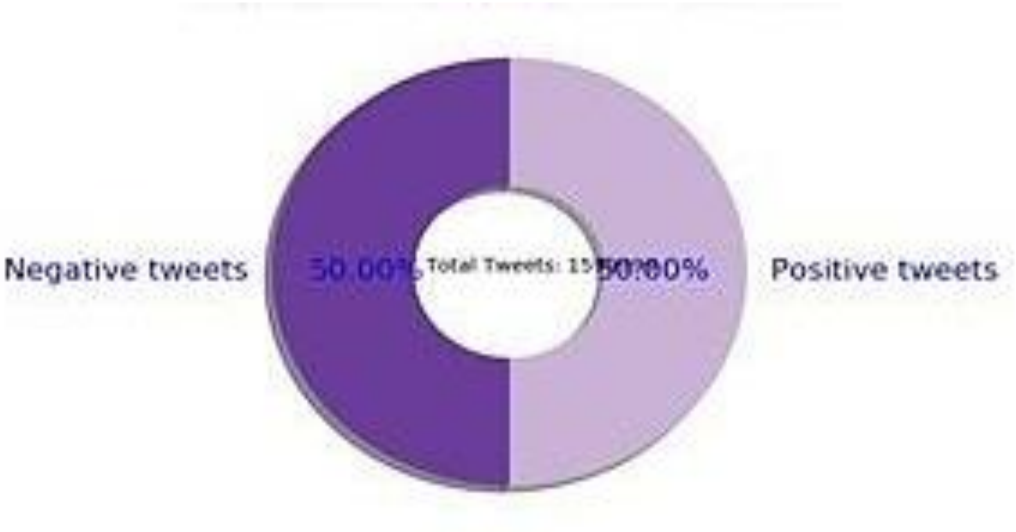


Figure (3.5) Description of dataset 4

3.4 Cleaning data

Given the fact that the tweets from the Twitter portal need a set of processors, the following pre-processing procedures have been adopted to normalize their content. This was done with the Keras package in Python with its own Tokenizer class:

- Remove punctuations.
- Lowercase and derived letters to reduce word conjugation.
- Remove stop words.
- Removes URLs, Hashtags, Mentions, Reserved words (retweet Favorite (RT-FAV)), Emojis, and Smileys.

There are also several tools available for cleaning Twitter data sets. These operations were performed NumPy, Pandas, Seaborn, Matplotlib, Basemap and word cloud on all data that was entered into the system in order to clean it and read the data properly.

3.5 Word Normalization

Stemming and Lemmatization are techniques for text normalization (or sometimes called term normalization), Languages have been spoken and written are made up of several words often derived from one another. When a language contains words that are derived from another word as their use in the speech changes is called inflected language natural language tool kit (NLTK) is a Python library to make programs that work with natural language. It provides a user-friendly interface to datasets that are over 50 corpora and lexical resources such as WordNet words repository. The library can perform different operations such as tokenizing, stemming, classification, parsing, tagging, and semantic reasoning Figure (3.6) shows example of stemming operation.

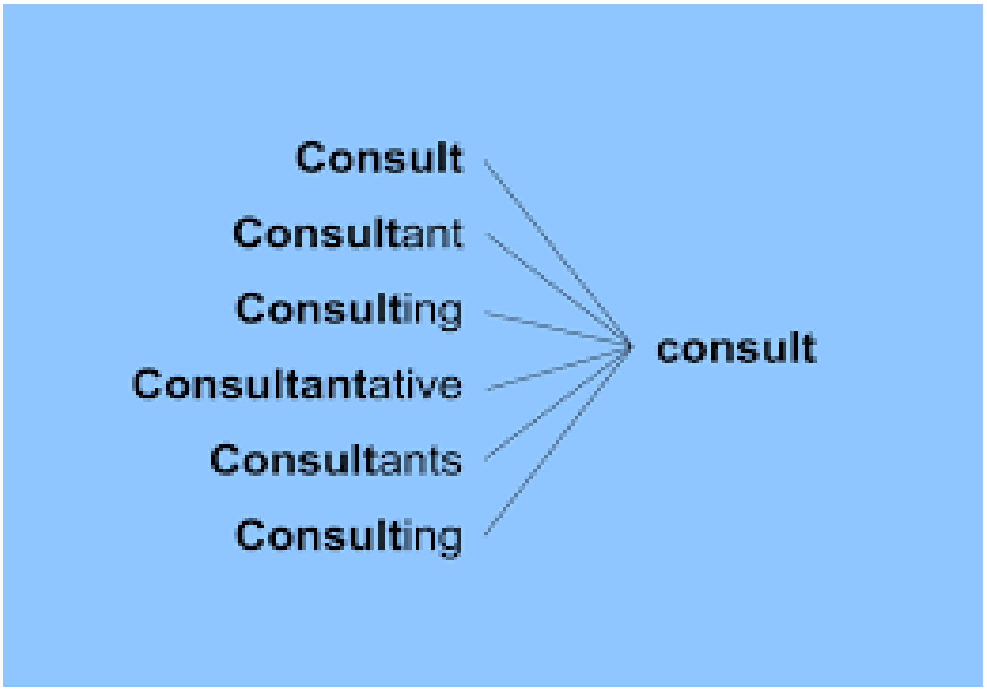


Figure (3.6) stemming operation

3.6 Shingling of data

The main objective of conducting the research is to classify similar tweets and evaluate the system according to the weight of each tweet. Therefore, the work has been prepared to improve the detection system for hate speech on social media and to make the result more accurate than the rest of the systems built in different way. Before matching tweets for similarity, they should be represented as groups. This process is called shingling, and it basically constructs the document based on N number of words to generate what is called a “**Token”.** Shingles for a sentence would be a set created from some consecutive words and then stored into a dictionary. For instance, table (3.1) shows example of two tweets and take k=3 for generate set of shingles

Table (3.1) generate shingles

<table>
<tr><th>Id</th><th>Tweets</th></tr>
<tr><td>1</td><td>Let's kill human and kill them for fun</td></tr>
<tr><td>2</td><td>I love eating out the home today</td></tr>
<tr><td></td><td></td></tr>
<tr><th colspan="2">Shingles</th></tr>
<tr><td colspan="2">Let's kill human
kill human kill
human kill them
kill them fun
I love eating
Love eating out
Eating out home
Out home today</td></tr>
</table>

3.7 Shingles Hashing

After shingling tweets and forming them into sets, the technique passes through another process called hashing in order to compact the data and reduce its size. We use in our work MD5 and CRC32 bit hashes as mention it in chapter (2) section (2.7) table (3.2) shows example of hash shingles

Table (3.2) Example of Hash function

shingle	MD5	CRC32
kill human kill	6d82f4139aad474dee0aa0e223f5f839	af37d865
human kill them	2a9c8bcd2fe8088276b3e87dcf650eea	bc74bccd
Kill them fun	a0a663b9e45ff8b0a65c3b18add4ed53	c1305207

3.8 Characteristic matrices

The tweets were set to smaller subgroups that were generated from the shingles and were fragmented. The next step in the technology process is to create a characteristic matrix, which represents each group or tweet as columns while indicating the items of standardized groups as rows. Generally, the cell is given a value of 1 when the shingles is part of the group, otherwise it is given a 0. The following Table (3.3) illustrates the method for representing shingles in the characteristic arrays

Table 3.3: Characteristic matrices for hashing K-shingle

Hash	**Tweets**			
	T1	T2	…	Tn
H1	1	0	…	1
H2	0	1	…	0
:	…	…	…	…
Hn	1	1	…	0

3.9 MinHash and signature matric

For more accurate results, the Minhash hash function was used. The Minhash is calculated in the document by specifying change rows for a column from the featured matrix described earlier. As can be seen, the quantity of Minhash for each column is equal to the number of the first row in the alternative request, where the column has a value of 1. The steps for creating the signature matrix are shown in table (3.4).

Table (3.4) generate signature matrix

Shingles	T1	T2	T3	T4	X+1 MOD 5	3X+1 MOD 5
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

Hash	T1	T2	T3	T4
H1	∞	∞	∞	∞
H2	∞	∞	∞	∞

(A)

Hash	T1	T2	T3	T4
H1	1	∞	∞	1
H2	1	∞	∞	1

(B)

Hash	T1	T2	T3	T4
H1	1	∞	2	1
H2	1	∞	4	1

(C)

Hash	T1	T2	T3	T4
H1	1	3	2	1
H2	1	2	4	1

(D)

Hash	T1	T2	T3	T4
H1	1	3	2	1
H2	0	2	0	1

(E)

Hash	T1	T2	T3	T4
H1	1	3	0	1
H2	0	2	0	0

(F)

Steps of tabels(A,B,C,D,E,F) explain how signature matric generated.

Consequently, the signature matrix consists of the hashes as rows and the tweets as columns. Each column refers to the Minhash signature for the tweet. It should be noticed that the signature matrix is much smaller than the characteristic matrix. The following table (3.5) shows the representation of the signature matrix.

Table (3.5) The representation of Signature matrix

Hash	**Tweets**			
	T1	T2	…	Tn
H1	0	5	…	3
H2	1	2	…	0
:	…	…	…	…
Hn	1	1	…	0

3.10 K-shingle Approaches

a. The k-shingle is K of the tokens (words) seen consecutively in a tweet. Occasionally, to slice short-length strings and use groups of hash values to convey tweets, hash shingles are useful. The K-shingle then behaves like the actions below:

b. The K words are chosen to divide the tweet into tokens.

c. The characteristic matrix is generated. If three tweets T1, T2, and T3 are used, then the tweet T1 consists of "I love this car because have red color", T2 consist of "I love this house" and T3 consist of "I love this woman" with k=3.

Table (3.6) shows the characteristic matrix that includes the tokens of shingles in column (1) and its existence in tweets. The other two tweets T2 and T3 are in the same process.

Table (3.6) Characteristic matric-based k-shingle

Shingles	**T1**	**T2**	**T3**
I love this	1	1	1
Love this car	1	0	0
This car because	1	0	0
Car because have	1	0	0
Because have red	1	0	0
Have red color	1	0	0
Love this house	0	1	0
Love this woman	0	0	1
Shingle m	m	m	m

Algorithm (3.1): The major steps of k-shingle hashes for Tweets

Algorithm (3.1): K-shingle Algorithm

Input	n tweets, k number of shingles
Output	Characteristic Matrix (M)

```
Begin
1. Preprocess the Tweets text by
     • removing the punctuation
     • removing adjusting the white space
2. Choose k number (n-k+1)/ k input by user its represent the length of shingles
3. Set tweets to group based on k
4. Hashing set (shingling) ax +b mod c Eq (2.1)
5. Find existing tokens in tweets
6. Generate characteristic Matrix (1 for exist hash shingle, 0 otherwise)
End
```

3.11 Minhash Technology

Minhash is a technique for approximating the Jaccard Similarity between two different set, making use of “random hash functions”. The aim of using Minhash is to exchange huge sets of data by smaller representations that are called signatures. The importance of using the Minhash property is that the signatures need to be compared for the two sets of data. Algorithm (3.2) shows the main steps of Minhash functions hashing on documents.

Algorithm (3.2): Minhash hashing functions

Input	**Characteristic Matrix M, Hash Functions h1, h2, h3, …, hn.**
Output	**Signature Matrix (S)**
Begin	
1.	Picking n randomly hashing functions h1, h2, h3, …, hn.
2.	Create the Matrix S signature from the Matrix M property, where each row I is a hash function and each column (c) is a tweet. Then set SIG (i, c) to have the hash h function and column c as the signature matrix part. Convert a vector of long bits into short signatures. Perform the following steps for each column c in the documents: • If c has 0 in the r lines of both texts, do nothing. • If the row has 1, set SIG (i, c) to the lower value of the current value of SIG (i, c) and hi(r) for each i=1,2, ..., n Then $Pr[h\pi(c1) = h\pi(c2)] = sim\ (c1, c2)$
END	

Supposing that there are 10 random hash functions (10 different combinations of a and b):

a. The first hash function is taken and applied to all the shingle values in the document.
b. Find the Minimum hash (Minimum hash is the name of hashing algorithm) to use it as the first ingredient of the Minhash signature.
c. Take the second hash and as before we should find the minimum hash value to use it as the second ingredient of the Minhash signature and so on to the N values we have.

3.12 TF-IDF Approach

TF*IDF is an information retrieval technique that weighs a term’s frequency (TF) and its inverse document frequency (IDF). Each word or term has its respective TF and IDF score. The product of the TF and IDF scores of a term is called the TF*IDF weight of that term. The TF*IDF algorithm is used to weigh a keyword in any content and assign the importance to that keyword based on the number of times it appears in the document. More importantly, it checks how relevant the keyword is throughout the web, which is referred to as corpus.

Algorithm (3.3) shows the main steps of TF-IDF hashing for documents.

```
Algorithm (3.3): TF-TDF Algorithm
Input     N tweets, signature matrix
Output    matrix contain max (TF-IDF)
Begin
1. Set k=number of shingles
2. Hashing tokens
3. Calculate TF // the number of times the token appeared in each document
Calculate IDF // number times this object appears in all document / number of document
4. Calculate TF-IDF $W_{s,set} = TF_{s,set} \log (N/DF_s)$  Eq (2.5)
End
```

Assume the tweets T1, T2, and T3:

T1 = “It is cold outside” T2= “The weather is cold” T3 = “I am outside”

The vocabulary developed using the three sentences listed above will be:

[it, is, cold, outside, the, weather, I, am]

To construct feature vectors from the sentence, this vocabulary of terms will be used. Let's see if they do it. For T1, the function vector will be:

T1= [1, 1, 1, 1, 0, 0, 0, 0]

T2 = [0, 1, 1, 0, 1, 1, 0, 0]

T3 = [0, 0, 0, 1, 0, 0, 1, 1]

In T1, for example, the TF for the word "outside" will be 1/4 = 0.25. Equally, the IDF in T1 would be Log (3/2) = 0.176 for the word 'outside'. This could be a TF-IDF value of 0.25 x 0.176 = 0.044.

3.13 Document to vector

There are multiple ways to change the text into vectors containing numerical values, such as:

a- Label Encoding
b- Custom Binary Encoding
c- One-Hot Encoding

But these standard methods lose the context of a given text. Word2vec attempts to solve this problem by creating vectors out of words; similarly, doc2vec is used to build vectors out of a document, independent of the document length. Doc2vec uses an unsupervised learning approach, the only difference from the model word to vector the parargaph_id, also known as a paragraph vector, was added to portray missing data from a document’s context، as mentioned in (2.10) there are two type of DOC2VEC (PV-DM and PV-DBOW) .

Algorithm (3.4): **DOC2VEC**

```
Input:
    • Input data
    • tagged _ document(list_of_list_of_words)
Output: matrix of vectors
Steps:
    1- training_data = list(tagged_document(data)).
    2- Initialize the models
        a- If dm=0 model PV-DM, if dm=1 model =PV-DBOW
        b- Set vector size
        c- Select the epochs number of iterations
    3- Building the vocabularyGet the greatest likelihood
         model_pvdm.build_vocab(training_data)
         model_dbow.build_vocab(training_data).
    4- Training the models
```

3.14 Classifier phase

Classification is a term used both for the classification method (distinguishing and distributing kinds of "things" into various groups) and for the resulting class collection, as well as for the assigning of items to pre-established classes. A basic idea and part of almost all kinds of operations is to classify data in the broad context described above, and it is an interdisciplinary area of analysis. Five classification techniques have been used in the proposed method, including Naive Bayes algorithm, Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), and Neural Network before enter the data to classifier the dataset divided into training (70%) and testing (30%) because of high rate in accuracy achieved in this splitting.

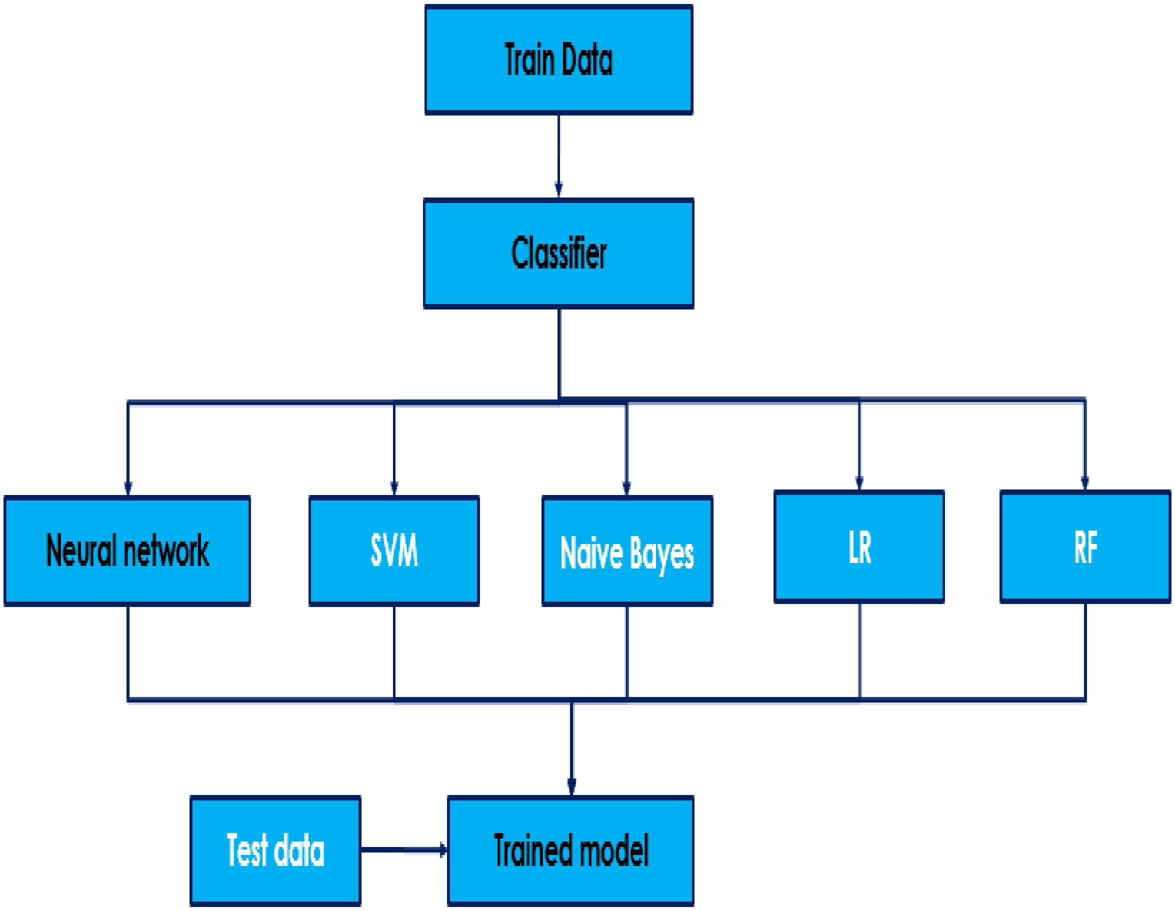


Figure (3.7) classifier phase

3.13.1 Naive Bayes algorithm

The training dataset input to this algorithm to build models as in the following stages:

Algorithm (3.5): Gaussian Naïve Bayes

```
Input:
    • Split the dataset into training (70%) and testing (30%)
    • Training dataset T. S, where S= set of Classified instances of tweets cases
    • F = ($f_{1,}$ $f_{2,}$ $f_{3,}$ ……, $f_n$)
Output: A class of testing dataset
Begin:
    Require: S ≠ Ø, num_attributes > 0
    Steps:
    5- Read the training dataset T.
    6- Calculate the mean and standard deviation of the predicator variables in each class.
    7- Repeat: Calculate the probability of $f_i$ using the gauss density equation (2.4) in each class
       Until the probability of all predictor variables ($f_{1,}$ $f_{2,}$ $f_{3,}$ ……, $f_n$) has been calculated
    8- Calculate likelihood for each class.
    9- Get the greatest likelihood.
end
```

3.13.2 Support Vector Machine ('SVM')

The training tweets data is input to SVM algorithm to predict hate speech tweets cases.

Algorithm (3.6): Linear SVM

```
Inputs
    • Split the dataset into training (70%) and testing (30%)
    • Training set {xi, $y_i$, $i$=1….. $L$}
Output
    Support vector machine models
Weight $\boldsymbol{w}$
Bias $b$
Begin:
        1. if $\boldsymbol{y = w^T x_i + b = 0}$ then
               $(\boldsymbol{x_i, y_i})$ is support vectors then save the parameters $\boldsymbol{w}$, $b$
        2. elseif $\boldsymbol{y = w^T x_i + b = 1}$ then
               save the parameters $\boldsymbol{w}$, $b$
        3. elseif $\boldsymbol{y = w^T x_i + b = -1}$ then
               update the parameters $\boldsymbol{w}$, $b$
End
```

3.13.3 Logistic Regression (LR)

The training tweets data is input to LR algorithm to predict hate speech tweets cases.

Algorithm (3.7): Linear LR

```
Input:
    • Split the dataset into training (70%) and testing (30%)
    • Training Algorithm L (Logistic Regression)
    • Sample Matrix X
    • Labels Vector y = [1….., K]
    • Initial Regressor Parameters Vector $\boldsymbol{\theta_i}$
Output:
  $\boldsymbol{\theta_i}$ Parameters vector for each regressor
Begin:
    For I=1: k
            Create a new binary vector $y_i$ for each label
             Where $y_i$ =1 if it belongs to the label and
             $y_i$=0 If it does belong.
             Appley L to X to find $\boldsymbol{\theta_i}$
End
```

3.13.4 Random Forest (RF)

The training tweets data is input to RF algorithm models for all cases of the tweet's groups and the output this algorithm is multiple decision tree as shown in algorithm (3.8):

Algorithm (3.8): RF

```
Input:
    • Split the dataset into training (70%) and testing (30%)
    • Training dataset T. S, where S= set of Classified instances of twitter cases
Output: majority votes to predict class for twitter case
Require: S ≠ Ø, num_attributes > 0
Begin:
    1- Assume number of twitter cases in the training set is N.
    2- Then, sample of these N cases is taken at random but with replacement.
    3- M input variables of features.
    4- A number m < M is specified such that at each node, m variables are selected
       at random out of the M.
    5- The best split of these m is used to split the node.
    6- The value of m is held constant while the forest grows.
    7- Each tree is grown to the largest extent possible and there is no pruning.
    8- Predict new data by aggregating the predictions of the n trees (i.e., majority
       votes for classification, average for regression).
End
```

3.13.5 Neural network

After convert dataset to set of vectors using document to vector models, and spilt to train and test set then use neural network classifier to train the model and evaluate the results. This model continues three hidden layers with input continue 300 neurons with function activation='relu' and output layers equal to the number of class with function activation='softmax'.

Algorithm (3.9): neural network

```
Input:
  Select num (input layers, hidden layers)
  Output layers= num of class of dataset
```

```
    Function(x;θ)parameterized with parameters θ.
    Training set of inputs x1…... $x_n$ and outputs y1…. $y_n$.
    Loss function L.
Output: training data
Begin:
    Stochastic Gradient Descent Training
        while stopping criteria not met do
                Sample a training example xi; $y_i$
                Compute the loss L (f ($x_i$; θ); $y_i$)
                $\widehat{\mathbf{g}}$ ← gradients of L (f ($x_i$; θ); $y_i$) w.r.t θ
                $\boldsymbol{\theta \leftarrow \theta + \eta_k \hat{g}}$
          return θ
****Computation Graph Forward Pass
for i = 1 to N do
Let $\boldsymbol{a_1, \dots, a_m = \pi^{-1}(i)}$
$\boldsymbol{v(i) \leftarrow f_i\left(v(a_1), \dots, v(a_m)\right)}$
******Computation Graph Backward Pass
$\boldsymbol{d(N) \leftarrow 1}$
for i = N-1 to 1 do
$\boldsymbol{d(i)}$ |← $\boldsymbol{\sum_{j \in \pi(i)} d(j) \cdot \frac{\partial f_j}{\partial i}}$
*****Neural Network Training with Computation Graph Abstraction
Define network parameters.
for iteration = 1 to N do
        for Training example xi; yi in dataset do
                loss node←build computation graph (xi, yi, parameters)
                loss node. Forward ()
                gradients ← loss node (). backward ()
```

```
7:              parameters ← update parameters (parameters, gradients)
8: return parameters.
```

3.14 Evaluation phase:

It is necessary to quantify the outcomes of the algorithm while developing a machine learning model. A dataset is usually split into a dataset for preparation and a dataset for research. For testing the proposed model, the testing dataset is used, while the test dataset is used to assess the model 's results.

Algorithm (3.10): Confusion Matrix (performance measure)

Input: Array of the obtained output form the proposed algorithms	
Output: Confusion Matrix	
Step1:	Begin
Step2:	start for loop to compromise all the pre-defined dataset For i =1 to k do

Step3:	Set TN←0, TP←0, FP←0, FN←0
	for i=0 toTestPoints.count
	for each (packet p inTestPoints[i])do
	if (p.actual=="normal") && (p.predicted=="normal")
	TN:=TN+1
	else if (p.actual=="normal") && (p.predicted=="hate")
	FP:=FP+1
	else if ((p.actual=="hate") && (p.predicted=="normal")
	FN:=FN+1
	else if ((p.actual=="hate") && (p.predicted=="hate")
	TP:=TP+1

	else
	p.predicted:=" outliers"

Step4:	Add TN, FP, FN, TP to (confusion matrix) CM
Step5:	End for
	End

3.15 summary

Overall, this chapter proposed the main concepts of the hate speech classification, dealing with data that represents a human report on social media like Twitter. As long as the reason of the whole project is to enhance the result that classifying hate speech on media network, so different techniques have to be used, which are K-shingle, TF-IDF, D2V, and the data mining classifier. The processing steps of the strategies are the same and show similar results, but in this chapter, the technique applied on the data set used to test the system and presented better results. The random forest classifier is one of the algorithms that have been explained in this chapter, along with two techniques. The algorithm was implemented with k-shingle and TF-IDF techniques, and their results were discussed in light of their output as compared to other approach. IT was shown that the proposed system provided the best results for word vector learning with neural network and k-shingles with random forest classifier. Finally, this chapter includes a number of evaluation metrics that are used to measure the enhancement the results in order to complete the comparison of the two techniques,doc2vec with neural network and Minhash with random forest classifier, after applying it to a real data set.

Chapter Four

System Implementation and Results Analysis

4.1 Introduction

This chapter presents a discussion of the experimental results for the proposed system, by using two different type of data sets that are label as hate speech, offensive language and normal speech. First, the data sets are preprocessed through approaches that were described Chapter Three., after which the data is represented in a form that is acceptable for machine learning algorithm classifiers to train the model. The ways that were used to represent data are Document-to-Vector and K-shingle with TF-IDF. Typically, Figure (4.1) shows a testing structure parameter of this chapter, as the implementation results of k-single and TF-IDF techniques will be shown onto tweet datasets. Then, the result of Minhash technique implementation on K-shingle and TF-IDF are presented on both data types. Finally, the classifier implementation results are proposed also for all datasets utilized with the system.

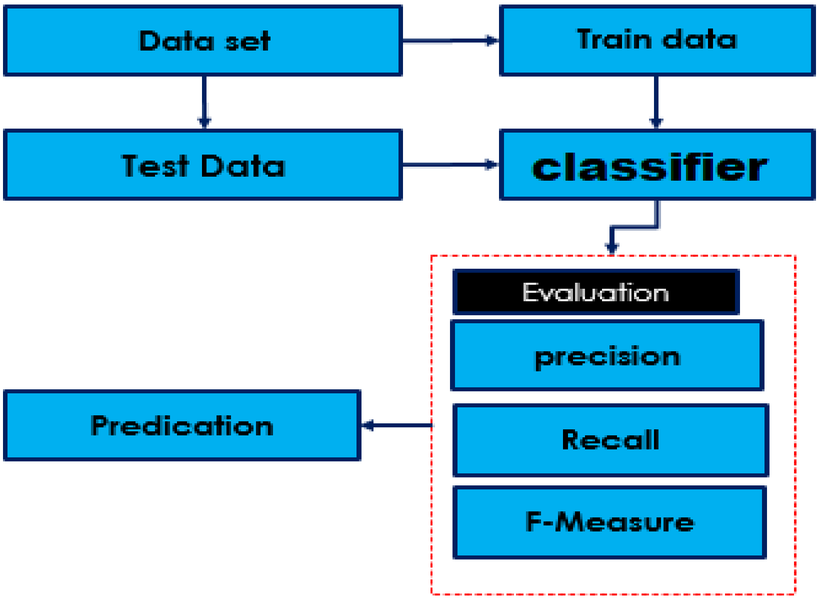


Figure (4.1) Testing model

4.2 Tweet Datasets

In short, datasets are csv file, with a content of plain texts, classes and the ID of each tweet. Each line of the content has a unique ID to refer to it. Typically, in this thesis there are groups of Twitter datasets used that consist of 31000,24000,39000, … etc. lines of content. This content is a plain text without link, characters, URL, video or audio. Table 4.1 shows a clip of dataset and represents what its content looks like.

Table (4.1) Dataset structure in a dataset

ID	Label	Tweet
0	2	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. &amp; as a man you should always take the trash out...
1	1	!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!
2	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confused as shit
3	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny
4	1	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told it to ya &#57361;
5	1	!!!!!!!!!!!!!!!!!!"@T_Madison_x: The shit just blows me..claim you so faithful and down for somebody but still fucking with hoes! &#128514;&#128514;&#128514;"
6	1	!!!!!!"@__BrighterDays: I can not just sit up and HATE on another bitch .. I got too much shit going on!"
7	1	!!!!&#8220;@selfiequeenbri: cause I'm tired of you big bitches coming for us skinny girls!!&#8221;
8	1	" &amp; you might not get ya bitch back &amp; thats that "
9	1	" @rhythmixx_ :hobbies include: fighting Mariam"bitch
10	1	" Keeks is a bitch she curves everyone " lol I walked into a conversation like this. Smh
11	1	" Murda Gang bitch its Gang Land "
12	1	" So hoes that smoke are losers ? " yea ... go on IG
13	1	" bad bitches is the only thing that i like "

4.2.1 Clean Dataset Results

The first stage of the initial processing process is cleaning the data by removing all non-alphanumeric characters, reducing word inflections with lowercase and stemming state, Twitter Mentions, URLs and Space Patter, as shown in Table (4.2).

Table (4.2): Clean Dataset

ID	Tweets
1.	woman complain cleaning house amp man always take trash
2.	boy dats cold tyga dwn bad cufin dat hoe st place
3.	dawg ever fuck bitch start cry confused shit
4.	lok like trany
5.	shit hear might true might faker bitch told ya
6.	shit blows claim faithful somebody stil fucking hoes
7.	sit hate another bitch got much shit going
8.	cause tired big bitches coming us skiny girls
9.	amp might get ya bitch back amp that’s
10.	hobies include fighting mariam bitch
11.	keks bitch curves everyone lol walked conversation like smh
12.	lok like trany
13.	shit hear might true might faker bitch told ya
14.	shit blows claim faithful somebody stil fucking hoes
15.	sit hate another bitch got much shit going
16.	cause tired big bitches coming us skiny girls
17.	amp might get ya bitch back amp that’s

4.2.2 Stemming and Lemmatization result

The second stage of the initial processing process is cleaning the Stemming and Lemmatization. Stemming is a technique used to extract the base form of the words by removing affixes from them. It is just like cutting down the branches of a tree to its stems. Table (4.3) show the dataset after stemming.

Table (4.3) Stemming and Lemmatization

ID	Tweets
1.	woman complain clean hous amp man alway take trash
2.	boy dat cold tyga dwn bad cufin dat hoe st place
3.	dawg ever fuck bitch start cri confus shit
4.	lok like trani
5.	shit hear might true might faker bitch told ya
6.	shit blow claim faith somebodi stil fuck hoe
7.	sit hate anoth bitch got much shit go
8.	caus tire big bitch come us skini girl
9.	amp might get ya bitch back amp that
10.	hobi includ fight mariam bitch
11.	kek bitch curv everyon lol walk convers like smh
12.	murda gang bitch gang land
13.	hoe smoke loser yea go ig
14.	bad bitch thing like
15.	woman complain clean hous amp man alway take trash

4.3 K-Shingle Results

In this section of testing, the K-shingle approach is implemented as shown in Chapter Three. Table (4.4) illustrates how text from the tweet dataset is used and the K-shingle is implemented for a number of tweets that starts with 100 and end with 1000.The k-shingle value considered is k=3…, k=5.

Table (4.4) K-shingle and Tweets based on K

K	Tweets	Shingles
3	The sky is blue and the sun is bright	The sky is sky is blue is blue and blue and the and the sun the sun is sun is bright
4	The sky is blue and the sun is bright	The sky is blue sky is blue and is blue and the blue and the sun and the sun is the sun is bright
5	The sky is blue and the sun is bright	The sky is blue and sky is blue and the is blue and the sun blue and the sun is and the sun is bright

The following Table (4.5) shows an example of generating hash using CRC32 for each shingle. The existing of the shingle in a tweet is represented by “1” and “0” if it does not exist in that tweet.

Table (4.5) Characteristic Matrix for 1000 Tweets based on K- Shingles

Hash values of shingles	T1	T2	T3	T4	T5	T6	T7	…	T1000
482828702	1	1	0	1	0	0	0	…	0
3872477633	1	0	0	0	0	0	0	…	0
797238198	1	0	0	0	0	0	0	…	0
1919870764	1	1	0	1	0	0	1	…	0
2154081544	1	0	0	1	0	0	1	…	1
2177772828	1	1	0	0	0	0	1	…	0

1779362483	1	1	1	1	1	1	0	…	1
3902709430	0	1	0	0	1	0	1	…	0

The next step is finding the Minhash based on the proposed approach referred to in the previous chapter.

4.4 K-shingle and Minhash Results

As it has been proposed earlier, the characteristic matrix is fairly huge and needs to be made dense before using it. Therefore, the Minhash technique is used to generate signature matrix as proposed in Chapter Three. Table (4.6) explains the result of implementing hash function-based k-shingle to characteristic matrix.

Table (4.6) Values of Characteristic Matrix with Minhash

Hash Values of K-Shingles	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**…**	**T100**	**H1**	**H2**	**H3**
482828702	1	1	0	1	0	0	0	…	0	82	11	20
3872477633	1	0	0	0	0	0	0	…	0	20	25	29
797238198	1	0	0	0	0	0	0	…	0	29	42	40
1919870764	1	1	0	1	0	0	1	…	0	0	34	35
2154081544	1	0	0	1	0	0	1	…	1	52	60	16
2177772828	1	1	0	0	0	0	1	…	0	16	22	27
1779362483	1	1	1	1	1	1	0	…	1	11	10	19
3902709430	0	1	0	0	1	0	1	…	0	18	25	29

Hence, from the characteristic matrix, a signature matrix is generated, and its size is less than the characteristic matrix. The matrix’s rows refer to the Minhash number while its columns refer to the tweets as explained in Table (4.7).

Table (4.7) Signature Matrix

Hash	T1	T2	T3	T4	T5	D6	D7	…	T1000
#1	10664	1445	529	11165	2075	5777	168	…	4167
#2	9889	2166	8526	8402	3111	320	2489	…	1772
#3	9114	2887	8789	5639	4147	7074	333	…	3854

The results show an example of the signature matrix using different numbers of hash functions.

4.5 TF-IDF Results and Labeling Result

4.5.1 TF-IDF and Minhash Result

The characteristic matrix of TF-IDF shingle will be the same in k-shingle, with hashing of tweet words with zero and one number to test these words existing in tweets. Table (4.8) shows the characteristic matric after updating with Minhash values based on TF-IDF shingle. The important part of the whole process is generating the signature matrix, and this is an essential part of this task. Therefore, it is done first.

Table (4.8) Characteristics Matrix for 100 tweets based on Shingles

hash values	T1	T2	T3	T4	T5	T6	T7	…	T100
1779362483	1	0	0	0	1	0	0	…	1
3902709430	1	1	0	0	1	0	0	…	1
3065274067	1	1	1	0	0	1	0	…	0
855400435	1	0	0	0	0	1	0	…	0
1360227904	1	0	0	0	0	0	0	…	0
3379038100	1	0	0	0	0	0	0	…	0
619488559	1	0	1	0	0	0	1	…	0
3769758244	1	0	1	0	0	1	1	…	0

3241152367	1	0	0	0	0	0	0	…	1
1663679033	1	0	0	1	0	1	0	…	0
1767525575	1	0	0	1	0	0	0	…	0

After that, calculate the rate of TF-Idf from signature matrix using TF-IDF technique, as shown in Table (4.9).

Table (4.9) The TF-IDF of 100 tweets.

Tweets	Max tf-idf
T1	0.000330317
T2	0.000612048
T3	0.000612048
T4	0.00047416
T5	0.001605119
T6	0.000612048
…	…
T100	0.00047416

4.5.2 Results of Labeling Process:

It is noticed from the proposed results that the TF-IDF Minhash technique works better. After creating a dataset of shingles and calculating the MinHash and TF-IDF for all datasets, each tweet wil be assigned a label depending on the value of shingle TI-IDF and MinHash. This is followed by creating a label for each shingle to classify them correctly, as presented in Table (4.10).

Table (4.10) Label Result Shingles.

Shingle values	Class
1779362483	0
3902709430	1
3065274067	1
855400435	2

1360227904	1
3379038100	0
619488559	1
3769758244	0
3241152367	1
1663679033	2
1767525575	1

4.6 Doucument to vector results

In tabel (4.11) and (4.12) show the reprsenation of tagged word in tweet (['violent', 'means', 'to', 'destroy', 'the','organization'])in PV-DM and PV-DBOW.

Table (4.11) model-pvdm of document to vector

```
  0.35694772 -0.10253964 0.39486504 -0.10129555 0.10594704 -0.23
  563926
   -0.2229983   0.17017114 0.0287006   0.14938888 -0.12965494 0.
  23368631
   -0.30536756 0.12788671 -0.561427   -0.00377469 -0.04005634 0.
  27649418
    0.01997015 0.01843208 0.25009045 -0.24862956 0.13262197 0.30
  03846
   -0.03805748 -0.2393072   0.27215642 0.07914103 0.07592767 0.1
  3353615
   -0.0530552   0.04801668 -0.02680592 0.20832661 0.34471107 0.2
  4245302
    0.25058004 0.1618371 -0.09241835 0.01635375]
```

Table (4.12) model_pv-dbow of document to vector

```
0.45199996 -0.15646784 0.3174646 -0.22232878 -0.10927293 -0.18907
53
 -0.29214934 0.4271567   0.20716298 -0.09680679 0.1720901 -0.0884
0349
 -0.06440508 0.13645375 -0.36956224 0.1576675   0.29343307 0.4282
4957
 -0.24662368 -0.13694254 0.40236777 -0.06448166 0.26060292 0.0597
9332
  0.02122489 -0.35315463 0.3560334   0.02929415 0.11327077 0.4159
7137
  0.0454339 -0.06022693 -0.312752   -0.06002218 0.3320964 -0.1223
506
  0.0274367 -0.22826265 0.28608498 0.04552925
```

4.7 Classifier and Confusion Matrix Results

This section presents the results of the classifiers used on the databases after the cleaning and k-shingle with MinHash operations, where the quality of the classifier classification is calculated depending on the parameters confusion matrix accuracy, precision recall, and F1-score. Table (4.13) shows the result of the classifier Random Forest on dataset 2 . In this classifier, the accuracy ratio has reached 98.9 and PR= 100, RE=88 and F1=93.

Table (4.13) Results of Random Forest Classifier with K-Shingle on Dataset2

	CLASS	PR	RE	F1-SCOR	SUPPORT
	0	0.99	1.00	0.99	53371
	1	1.00	0.82	0.90	2536
	2	1.00	0.80	0.89	822
Accuracy				0.99	56729
Macro Avg		1.00	0.88	0.93	56729
Weighted Avg		0.99	0. 99	0. 99	56729
Accuracy Over All	0.98926				

Table (4.14) shows the results of Random Forest classifier with k-shingle on dataset 3 using k-shingle with Minhash. The obtained accuracy rate was 98.8 with PR= 99, RE=84 and F1-score =90.

Table (4.14) Result of Random Forest Classifier with K-Shingle on Dataset 3

	CLASS	PR	RE	F1-SCORE	SUPPORT
	0	0.99	1.00	0.99	5203
	1	1.00	0.68	0.81	188
Accuracy				0.99	5391
Macro Avg		0.99	0.84	0.90	5391
Weighted Avg		0.99	0. 99	0. 99	5391
Accuracy Over All	0.98887				

The results in Table (4.15) show the strength of the hash technique with many classifiers, as it shows the accuracy in using three types of works as shown below.

Table (4.15) Result of Three kind of Classifier

Classifiers	Precision	Recall	F1-Score	Accuracy
First dataset				
Rand. Forest	0.90	0.91	0.91	٠,٩١٤٦
SVM	٠,٩٠	٠,٩٢	٠,٩٢	٠,٩١٦٨
LR	٠,٩٠	٠,٩٢	٠,٩١	٠,٩١٢٩
Second dataset				
Rand. Forest	١,٠٠	**0.88**	**0.93**	٠,٩٨٩٢٦
SVM	٠,٩٨	٠,٨٩	٠,٩٣	٠,٩٤,٦
LR	٠,٩٨	٠,٨٩	٠,٩٣	٠,٩٤,٥
Third dataset				
Rand. Forest	٠,٩٩	٠,٨٤	٠,٩٠	٠,٩٨٨٨
SVM	٠,٨٩	٠,٧٠	٠,٧٦	٠,٩٥٦٣
LR	٠,٩٠	٠,٦٧	٠,٧٣	٠,٩٥٤٠

Figure (4.2) shows the result of stander dataset (1-2-3) without using Minhash and TF-IDF Figure (4.3) indicates that the proposed system and methods used have achieved very high results that exceed the results of previous studies. This implies the efficiency of the suggested system and the work of the k-shingles with Minhash techniques.

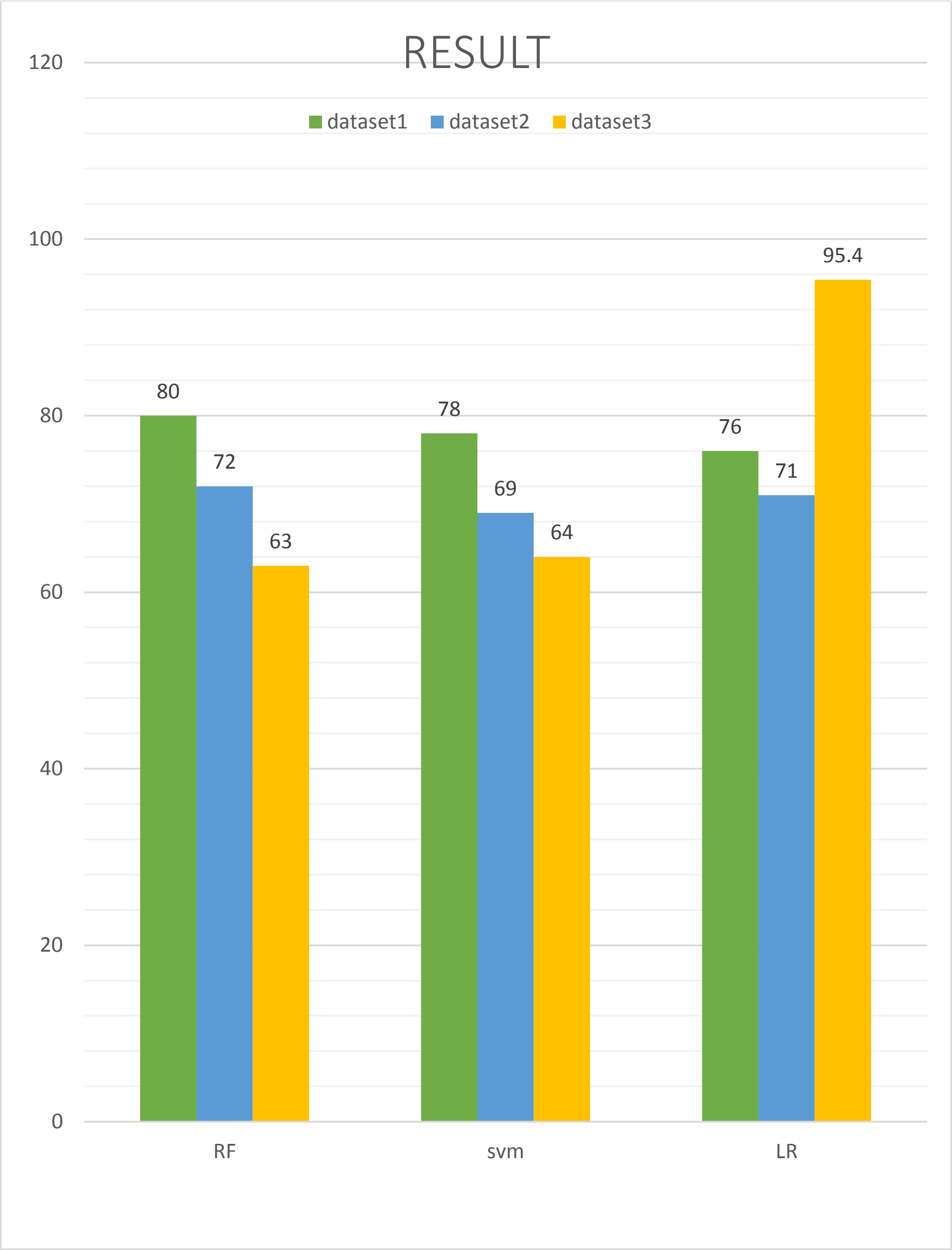


Figure (4.2) Result of stander datasets

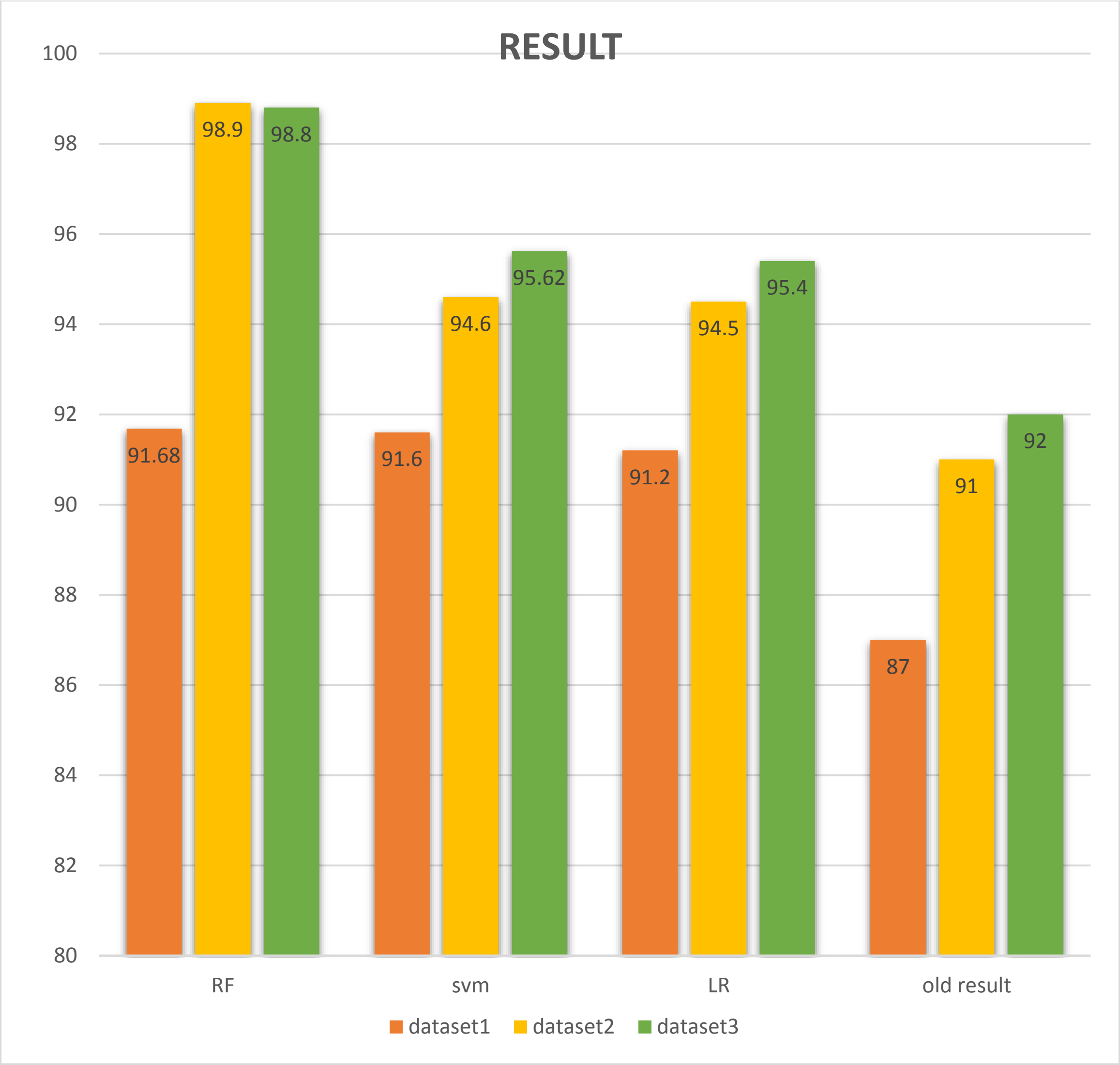


Figure (4.3) Comparing Obtained Result with Outcomes of Other Studies

The results of previous years for the same databases used in the thesis, compared to the results of our system are shown in Table (4.16).

Table (4.16) Comparison between results of the proposed model and alternative studies

Data set	Author(s)	Year	Accuracy	Accuracy of our system
Dataset 1	Watanabe, Hajime Bouazizi, Mond her Ohtsuki, Tomoaki	٢٠١٨	٠,٨٧٤	٠,٩١٦٨
Dataset 2	Davidson, Thomas Warmsley, DanaMacy, Michael Weber,	٢٠١٧	٠,٩١	٠,٩٨٩٢٦

	Ingmar			
	Watanabe, Hajime Bouazizi, Mondher Ohtsuki, Tomoaki	٢٠١٨	٠,٨٧	٠,٩٨٩٢٦
Dataset 3	MacAvaney, SeanYao, Hao Ren Yang, Eugene Russell, Atina Goharian, Nazli Frieder, Ophir	٢٠١٩	٠,٩٢	٠,٩٨٨٨

The second aspect of the present after the initial processing of the data is the process of converting it into vector, using document to vector technology, and entering data into the data mining classifiers. Table (4.17) shows the result of using D2V with RF classifier on dataset1.

Table (4.17) RF With D2V Dataset1

	Class	Precision	Recall	F1-Scor	Support
	0.0	0.94	1.00	0.97	5949
	1.0	0.75	0.14	0.23	444
Accuracy				0.94	6393
Macro Avg		0.84	0.57	0.60	6393
Weighted Avg		0.93	0.94	0.92	6393
Accuracy Overall	0.936805				

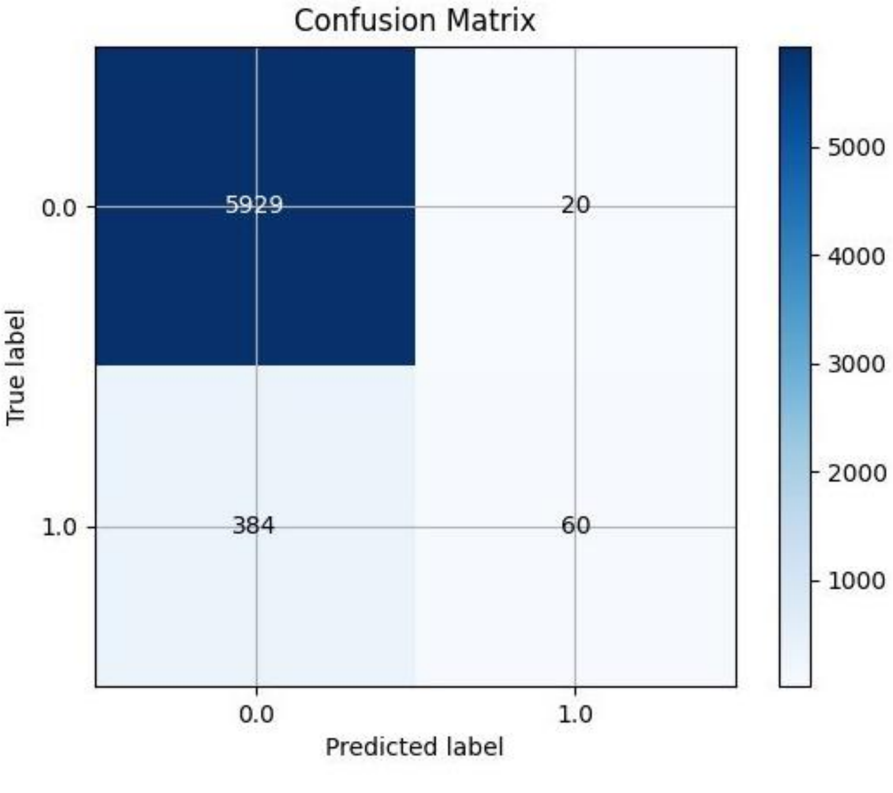


Figure (4.4) Confusion Matrix of RF Classifier

Figure (4.5) explain the plotting of confusion matrix that continues the true label and predicted label for data set in order to calculate the preference parameter. Table (4.18) illustrates the result of using document to vector with Logistic regression classifier.

Table (4.18) LR with D2V dataset1

	Class	Precision	Recall	F1-scor	Support
	0.0	0.93	1.00	0.96	5949
	1.0	0.39	0.04	0.07	444
Accuracy				0.93	6393
Macro avg		0.66	0.52	0.51	6393
Weighted avg		0.89	0.93	0.90	6393
Accuracy over	0.92956				

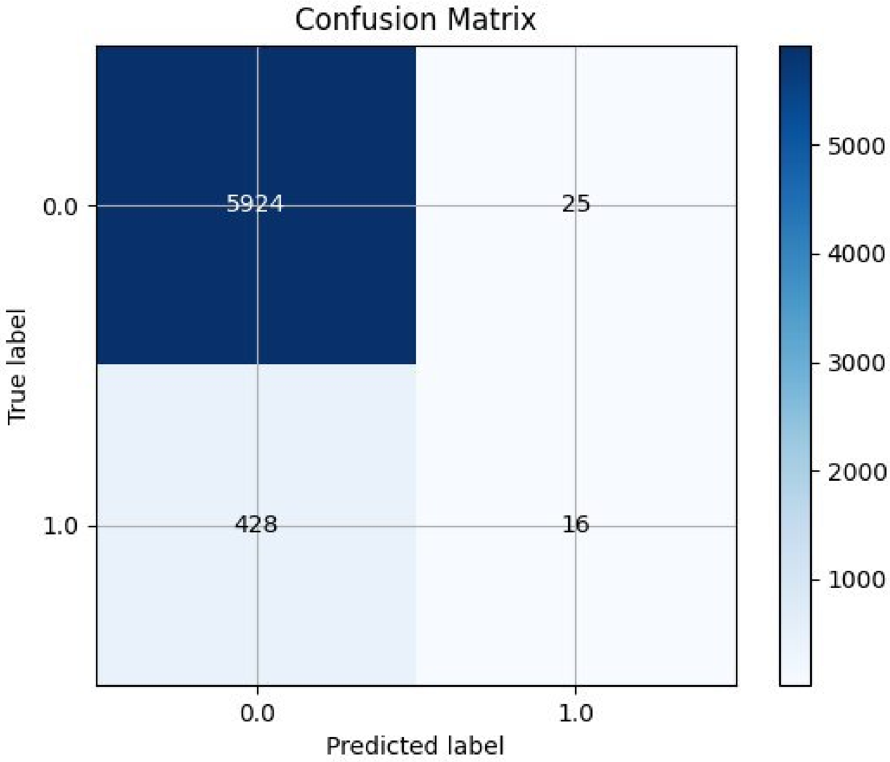


Figure (4.5) Confusion Matrix of LR Classifier

Another classifier used it to test the system is the Naive Bayes. Table (4.19) shows the result of preference parameter. In this classifier, the accuracy reached a percentage of 93.1.

Table (4.19) Naive Bayes Results

	Class	Precision	Recall	F1-scor	Support
	0.0	0.93	1.00	0.96	5949
	1.0	0.74	0.03	0.06	444
Accuracy				0.93	6393
Macro Avg		0.83	0.52	0.51	6393
Weighted Avg		0.92	0.93	0.90	6393
Accuracy Over	0.9319				

The results illustrated in Tables (4.20 and 4.21) show the result of neural network with 3 hidden layers and 20 epochs to train the data for obtaining best result. First, 64 patch sizes are taken from the train data, for which the accuracy rates are collected. Overall, collecting the accuracy and taking the average for all systems is performed by means of soft max layer. The accuracy rate for the neural network reached (93.8).

Table (4.20) Neural Network Result

Epoch 1/20
64/20455 [..............................] - ETA: 35s - loss: 0.6932 - accuracy: 0.4531
1088/20455 [>.............................] - ETA: 2s - loss: 0.5714 - accuracy: 0.9035
2048/20455 [==>...........................] - ETA: 1s - loss: 0.4694 - accuracy: 0.9141
2816/20455 [===>..........................] - ETA: 1s - loss: 0.4242 - accuracy: 0.9151
3584/20455 [====>.........................] - ETA: 1s - loss: 0.3866 - accuracy: 0.9199
4416/20455 [=====>........................] - ETA: 1s - loss: 0.3694 - accuracy: 0.9187
5184/20455 [======>.......................] - ETA: 1s - loss: 0.3541 - accuracy: 0.9198
6400/20455 [========>.....................] - ETA: 1s - loss: 0.3282 - accuracy: 0.9245
7232/20455 [=========>....................] - ETA: 0s - loss: 0.3249 - accuracy: 0.9231
8128/20455 [==========>...................] - ETA: 0s - loss: 0.3168 - accuracy: 0.9240
8896/20455 [============>.................] - ETA: 0s - loss: 0.3106 - accuracy: 0.9248
9664/20455 [=============>................] - ETA: 0s - loss: 0.3048 - accuracy: 0.9257
10432/20455 [==============>...............] - ETA: 0s - loss: 0.3018 - accuracy: 0.9257
11200/20455 [===============>..............] - ETA: 0s - loss: 0.3008 - accuracy: 0.9251
11968/20455 [================>.............] - ETA: 0s - loss: 0.2983 - accuracy: 0.9252

12736/20455 [=================>............] - ETA: 0s - loss: 0.2936 - accuracy: 0.9263
13440/20455 [==================>...........] - ETA: 0s - loss: 0.2903 - accuracy: 0.9269
14208/20455 [===================>..........] - ETA: 0s - loss: 0.2897 - accuracy: 0.9266
14976/20455 [====================>.........] - ETA: 0s - loss: 0.2870 - accuracy: 0.9271
15744/20455 [======================>.......] - ETA: 0s - loss: 0.2857 - accuracy: 0.9271
16448/20455 [=======================>......] - ETA: 0s - loss: 0.2846 - accuracy: 0.9271
17344/20455 [========================>.....] - ETA: 0s - loss: 0.2829 - accuracy: 0.9273
18368/20455 [=========================>....] - ETA: 0s - loss: 0.2797 - accuracy: 0.9280
19136/20455 [===========================>..] - ETA: 0s - loss: 0.2776 - accuracy: 0.9285
19840/20455 [============================>.] - ETA: 0s - loss: 0.2760 - accuracy: 0.9288
20455/20455 [==============================] - 1s 69us/step - loss: 0.2757 - accuracy: 0.9287

Table (4.21) Neural Network Results 2

Epoch 1/20
64/20455 [..............................] - ETA: 35s - loss: 0.6932 - accuracy: 0.4531
1088/20455 [>.............................] - ETA: 2s - loss: 0.5714 - accuracy: 0.9035
2048/20455 [==>...........................] - ETA: 1s - loss: 0.4694 - accuracy: 0.9541
2816/20455 [===>..........................] - ETA: 1s - loss: 0.4242 - accuracy: 0.9351
3584/20455 [====>.........................] - ETA: 1s - loss: 0.3866 - accuracy: 0.9499
4416/20455 [=====>........................] - ETA: 1s - loss: 0.3694 - accuracy: 0.9487
5184/20455 [======>.......................] - ETA: 1s - loss: 0.3541 - accuracy: 0.9198
6400/20455 [========>.....................] - ETA: 1s - loss: 0.3282 - accuracy: 0.9245
7232/20455 [=========>....................] - ETA: 0s - loss: 0.3249 - accuracy: 0.9231
8128/20455 [==========>...................] - ETA: 0s - loss: 0.3168 - accuracy: 0.9240
8896/20455 [============>.................] - ETA: 0s - loss: 0.3106 - accuracy: 0.9648
9664/20455 [=============>................] - ETA: 0s - loss: 0.3048 - accuracy: 0.9257
10432/20455 [==============>...............] - ETA: 0s - loss: 0.3018 - accuracy: 0.95257
11200/20455 [===============>..............] - ETA: 0s - loss: 0.3008 - accuracy: 0.9251
11968/20455 [================>.............] - ETA: 0s - loss: 0.2983 - accuracy: 0.9852
12736/20455 [=================>............] - ETA: 0s - loss: 0.2936 - accuracy: 0.9263
13440/20455 [==================>...........] - ETA: 0s - loss: 0.2903 - accuracy: 0.9969
14208/20455 [===================>..........] - ETA: 0s - loss: 0.2897 - accuracy: 0.9266
14976/20455 [====================>.........] - ETA: 0s - loss: 0.2870 - accuracy: 0.9271
15744/20455 [======================>.......] - ETA: 0s - loss: 0.2857 - accuracy: 0.9771
16448/20455 [=======================>......] - ETA: 0s - loss: 0.2846 - accuracy: 0.9271
17344/20455 [========================>.....] - ETA: 0s - loss: 0.2829 - accuracy: 0.9873

```
18368/20455 [========================>....] - ETA: 0s - loss: 0.2797 - accuracy: 0.9280
19136/20455 [==========================>..] - ETA: 0s - loss: 0.2776 - accuracy: 0.9485
19840/20455 [===========================>.] - ETA: 0s - loss: 0.2760 - accuracy: 0.9488
 20455/20455 [==============================] - 1s 69us/step - loss: 0.2757 - accuracy:
0.9687
```

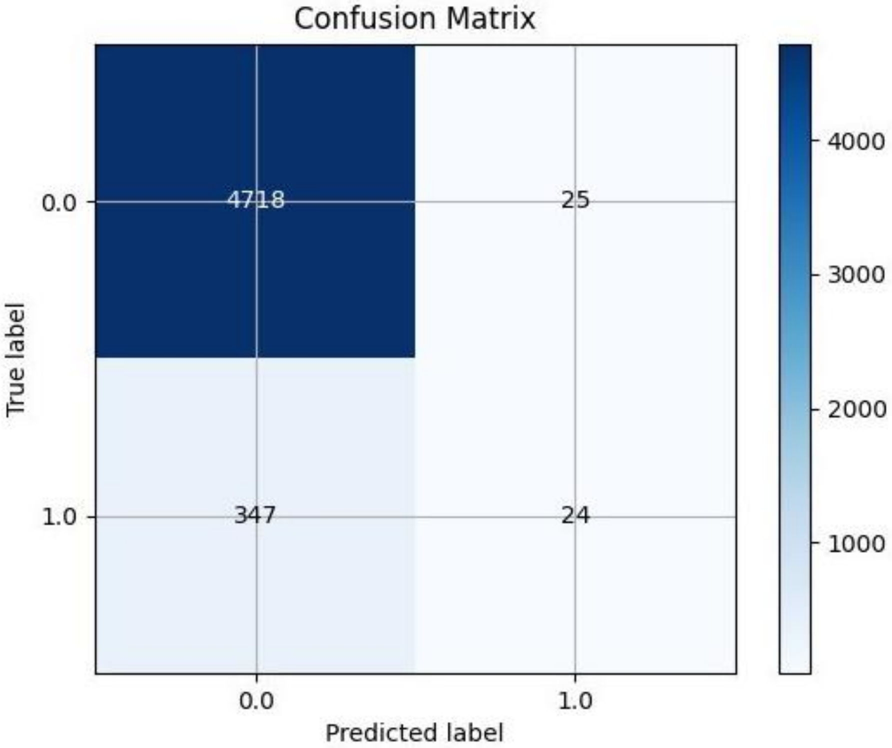


Figure (4.6) Confusion Matrix of Neural Network Classifier

Table (4.22) shows the result of dataset1 and dataset4 using model document to vector with data mining classifier.

Table (4.22) Results of Dataset1 and Dataset4 for all classifiers.

Classifiers	Accuracy
Dataset (1)	
Neural network	**93.8**
RF	93.6
Naive Bayes	93.19
LR	92.9
Dataset (4)	
Neural network	**٠,٩٥**
RF	84
LR	82.4
Naive Bayes	83.2

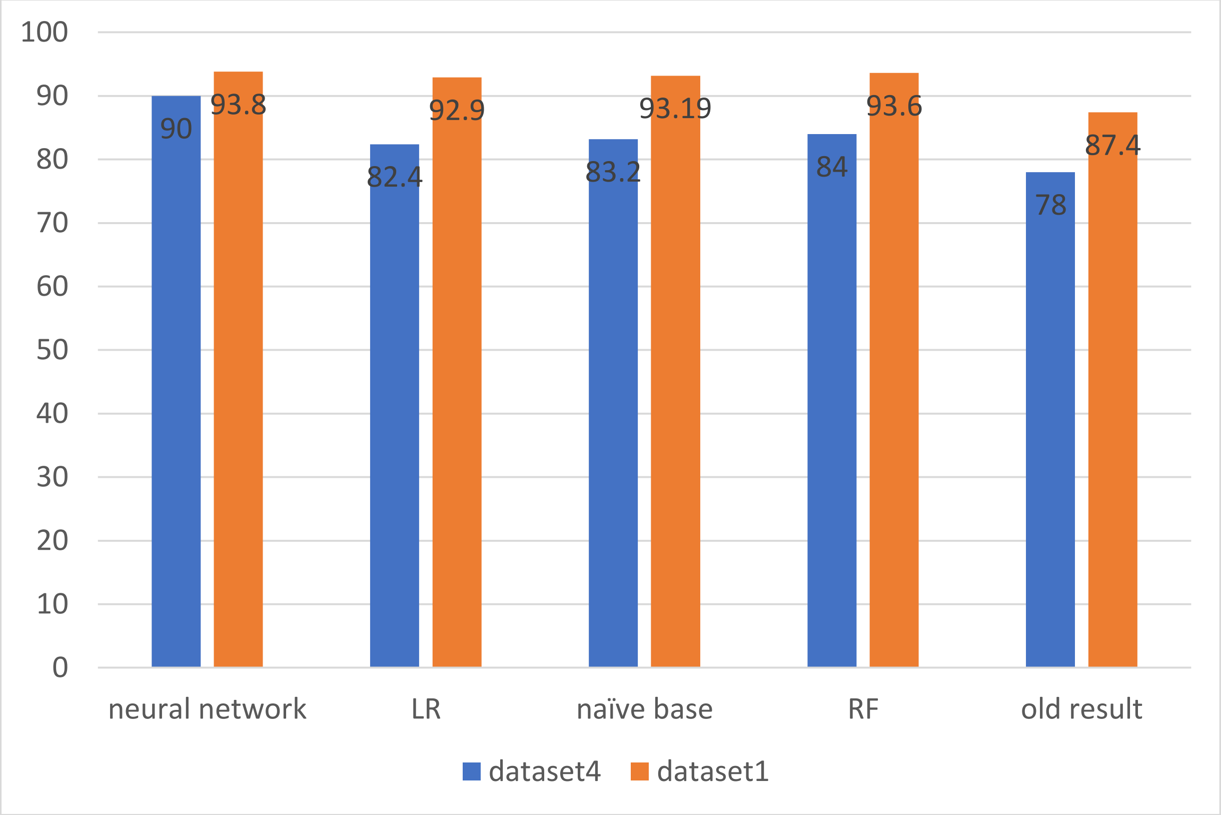


Figure (4.7) Accuracy Results of Dataset1 and Dataset4

Table 4.23: A comparison of results between the proposed model and alternative models

Data set	Author	Year	Accurac y	Accuracy of our system
Dataset 4	Kumar, Srivastava, Soni, Tyaghi, Singh	2020	0.78	0.90
Dataset 1	Watanabe, Hajime Bouazizi, Mondher Ohtsuki, Tomoaki	٢٠١٨	٠,٨٧٤	٠,٩٣٨

4.8 Conclusion

Overall, four data set have been used, each of which contains a class of each tweet to train the proposed model. Furthermore, many types of data preprocessing techniques have been implemented to obtain the best result when using it as input for the classifier. After that, the k-shingle, Minhash, TF-IDF, and D2V have been used. These techniques have been applied for a number of different datasets from social media so as to find the expected results with each technique. The data mining classifier was used to train the model (SVM-RF-LR-Naive Bayes- neural network). Moreover, four enhancing performance metrics have been employed for this thesis which are the accuracy, recall, f1-score, and precision.

Chapter Five

Conclusions and Future Works

5.1 Conclusion

In conclusion, this thesis has introduced the proposal of leveraging social media network for hate speech classification, whereby the following conclusions have been obtained:

1. K-shingles and TF-IDF help to recode data set into actual class assignment which lead to a high accuracy results compared with stander and related works**.**
2. Minhash technique has increased the quality of the whole work, it used to distribute the tweets in iterative process to generate characteristics matrix and signature matrix.
3. Document to vector an unsupervised learning algorithm that learns vector representations for variable length pieces of texts such as tweets, massage. The vector representations are learned to predict the surrounding words in contexts sampled from the paragraph.
4. Document to vector (doc2vec) I used in a proactive way to represent data as a vector without loss word order information, this model has achieved ahigh result with neural network.
5. When compared with the original data set, we find a very clear difference in the increase in the accuracy of classification when using data mining classifier with feature selection model like document to vector and Minhash algorithm.
6. Data mining classifier (SVM – LR – RF – naive Bayes - neural network) are chosen because have a higher accuracy, in comparison with other classifier that used it in related work.

5.2 Future work

The possible future works related to this project can be summarized in four points:

1. This model can be used to predict hate speech within an online data stream on social media networks.
2. An integration can be made between Minhash techniques and neural network to get more efficient results.
3. There are a number of categorized tweets, especially in the classification of hate speech. Therefore, an error analysis may help to provide feedback about the model 's efficiency. For example, this may help explain why this class is so difficult to predict while analyzing the hate class's inaccurate predictions. It would be important to see whether and how these words are helpful for discriminating between hate speech and offensive language.
4. complete, production-quality classifier will incorporate many different features beyond the vectors corresponding to the words in the text .meaning that add feature like location of user , age of user and node connected link to other user this may be help us to make it feature selection to increase the quality of model

List of Abbreviations	
LRC	Logistic Regression classifier
TF-IDF	Term Frequency and Inverse Document Frequency
RF	Random forest classifier
NLP	Neuro-linguistic programming
PR	Precision
SVM	Support Vector Machine
AC	Accuracy
D2V	Document to vector
W2V	Word to vector
PV-DM	Distributed Memory Architecture of Paragraph Vectors
SGD	Stochastic gradient descent
OSN	Online social network
NSA	National-Security Agency
RDF	Resource Description Framework
MD5	Message Digest 5
SHA	Secure Hash Algorithm
CRC	cyclic redundancy check
PV-DBOW	Bag of Words version of Paragraph Vector
NB	Naive Bayes
NLTK	Natural Language Tool Kit

Table of Contents

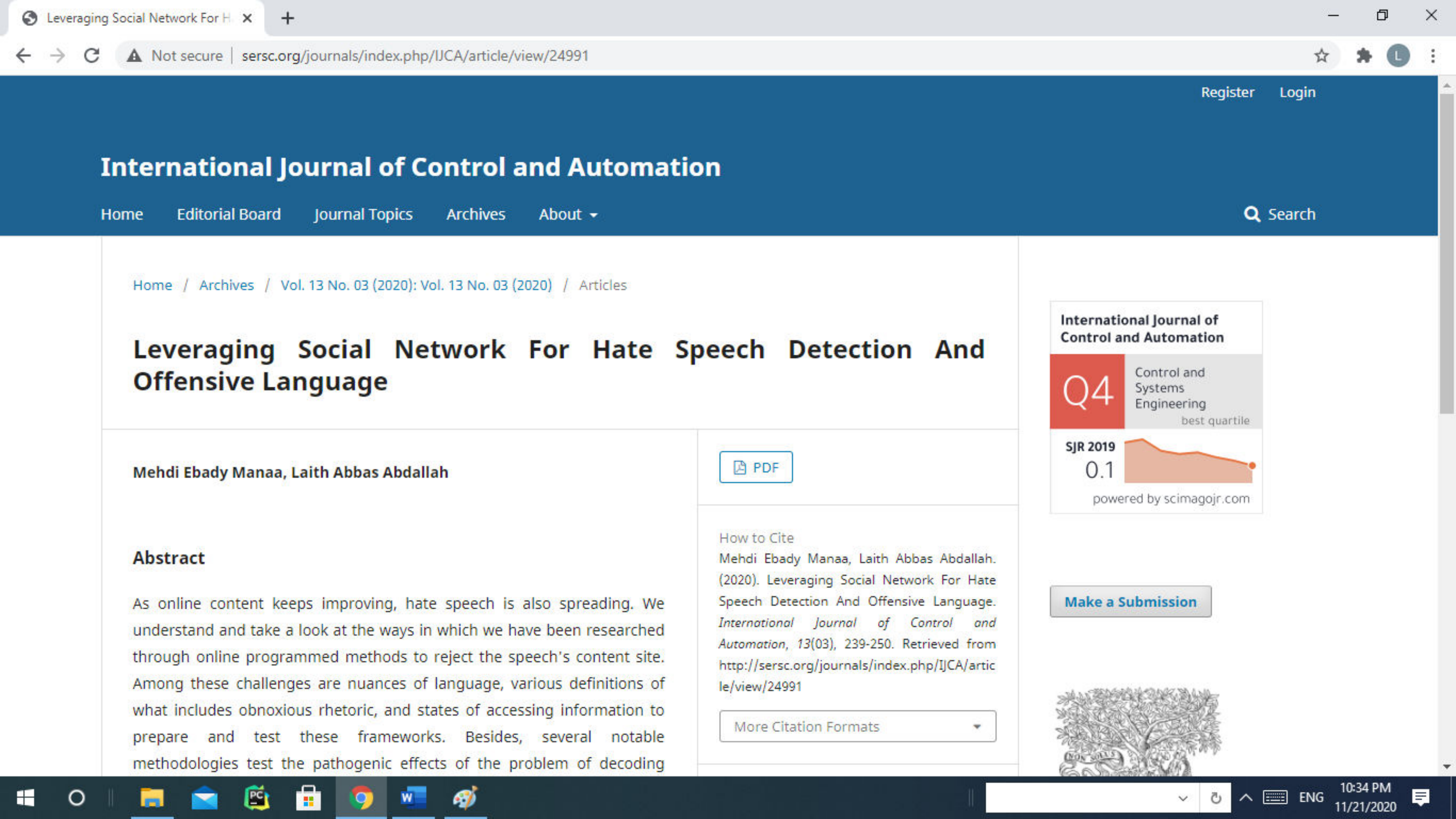
List OF Figures

List of Abbreviations

LRC	Logistic Regression classifier
TF-IDF	Term Frequency and Inverse Document Frequency
RF	Random forest classifier
NLP	*Neuro-linguistic programming*
PR	Precision
SVM	Support Vector Machine
AC	Accuracy
D2V	Document to vector
W2V	Word to vector
PV-DM	Distributed Memory Architecture of Paragraph Vectors
SGD	Stochastic gradient descent
OSN	Online social network
NSA	National-Security Agency
RDF	Resource Description Framework
MD5	Message Digest 5
SHA	Secure Hash Algorithm
CRC	cyclic redundancy check
PV-DBOW	Bag of Words version of Paragraph Vector

NB	Naive Bayes
NLTK	Natural Language Tool Kit

List OF Figures


Leveraging Social Network For H
Not secure
sersc.org/journals/index.php/IJCA/article/view/24991
Register
Login
Home
Editorial Board
Journal Topics
Archives
About
Search
Home / Archives / Vol. 13 No. 03 (2020): Vol. 13 No. 03 (2020) / Articles
PDF
International Journal of Control and Automation
Q4
Control and Systems Engineering
best quartile
SJR 2019
0.1
powered by scimagojr.com
Make a Submission
More Citation Formats
ENG
10:34 PM
11/21/2020

International Journal of Control and Automation

Leveraging Social Network For Hate Speech Detection And Offensive Language

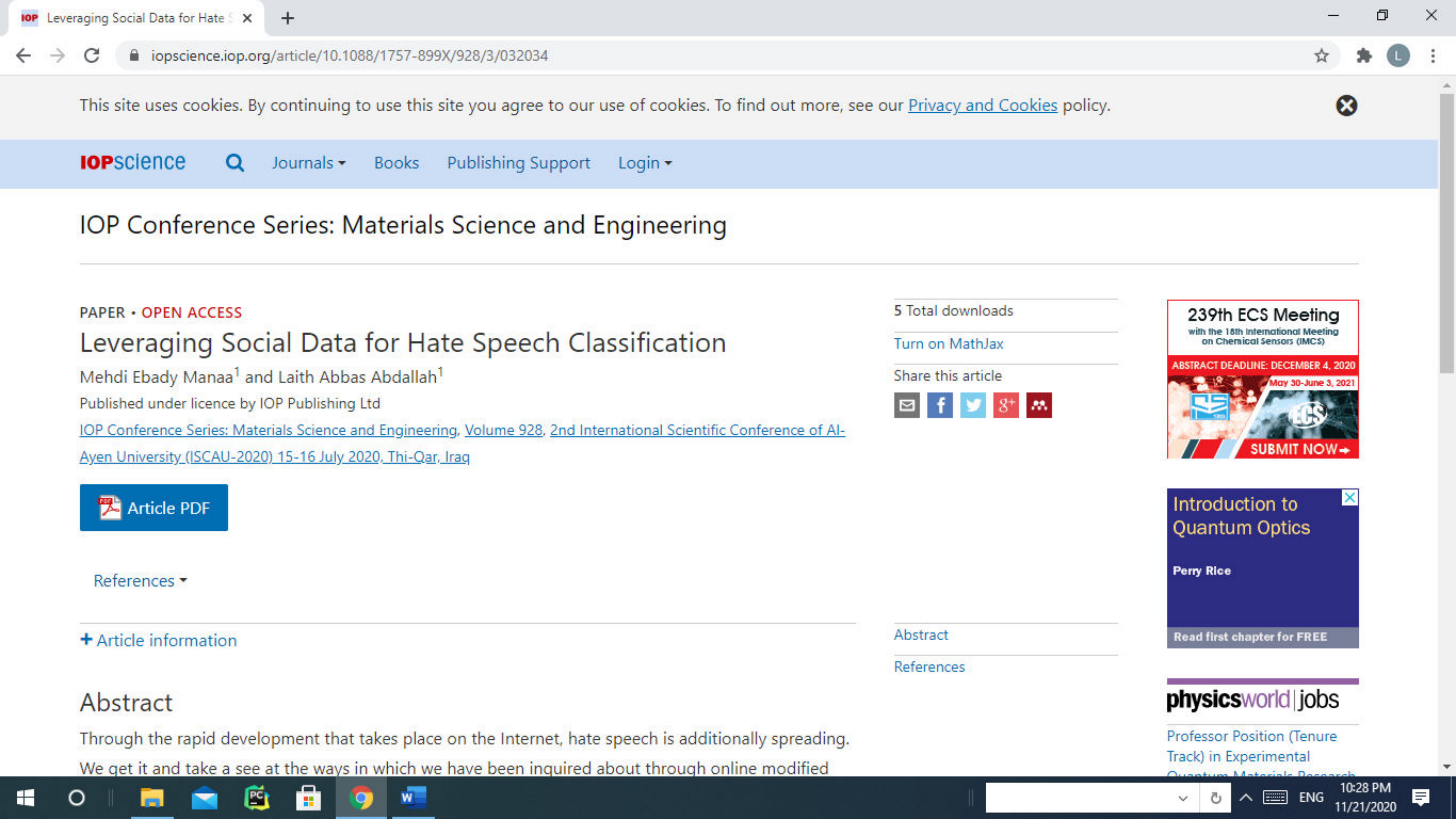
Mehdi Ebady Manaa, Laith Abbas Abdallah

Abstract

As online content keeps improving, hate speech is also spreading. We understand and take a look at the ways in which we have been researched through online programmed methods to reject the speech's content site. Among these challenges are nuances of language, various definitions of what includes obnoxious rhetoric, and states of accessing information to prepare and test these frameworks. Besides, several notable methodologies test the pathogenic effects of the problem of decoding

How to Cite

Mehdi Ebady Manaa, Laith Abbas Abdallah. (2020). Leveraging Social Network For Hate Speech Detection And Offensive Language. *International Journal of Control and Automation*, *13*(03), 239-250. Retrieved from http://sersc.org/journals/index.php/IJCA/article/view/24991


Leveraging Social Data for Hate S
iopscience.iop.org/article/10.1088/1757-899X/928/3/032034

IOPscience
Journals
Books
Publishing Support
Login

IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

Leveraging Social Data for Hate Speech Classification

Mehdi Ebady Manaa[1] and Laith Abbas Abdallah[1]

Published under licence by IOP Publishing Ltd
IOP Conference Series: Materials Science and Engineering, Volume 928, 2nd International Scientific Conference of Al-Ayen University (ISCAU-2020) 15-16 July 2020, Thi-Qar, Iraq

Abstract

Through the rapid development that takes place on the Internet, hate speech is additionally spreading. We get it and take a see at the ways in which we have been inquired about through online modified

REFERENCES

1- Zhang, Ziqi, and Lei Luo. "Hate speech detection: A solved problem? the challenging case of long tail on twitter." Semantic Web 10.5 (2019): 925-945.

2- B. T. N. Y. Times, “Christchurch Shooting Live Updates: 49 Are Dead After 2 Mosques Are Hit,” https://www.nytimes.com/2019/03/14/world/asia/new-zealand-shooting-updates-christchurch.html, 2019.

3- B. T. N. Y. Times, “Christchurch Shooting Live Updates: 49 Are Dead After 2 Mosques Are Hit,” https://www.nytimes.com/2019/03/14/world/asia/new-zealand-shooting-updates-christchurch.html, 2019.

4- Hate Speech—ABA Legal Fact Check—American Bar Association; Available from: https://abalegalfactcheck.com/articles/hate-speech.html.

5- Community Standards; Available from: https://www.facebook.com/communitystandards/objectionable content.

6- M. Mondal, L. A. Silva, and F. Benevento, “A measurement study of hate speech in social media,” in Proceedings of the 28th ACM Conference on Hypertext and Social Media, 2017, pp. 85–94, doi: 10.1145/3078714.3078723.

7- C. Zaghi, “Automatic detection of hate speech in social media,” no. June, 2019.

8- Atte Oksanen, James Hawdon, Emma Holkeri, Matti Näsi, Pekka Räsänen (2014), "Exposure to Online Hate among Young Social Media Users", in M. Nicole Warehime (ed.) Soul of Society: A Focus on the Lives of Children &Youth (Sociological Studies of Children and Youth, Volume 18) Emerald Group Publishing Limited, pp. 253–273.https://doi.org/10.1108/S1537-466120140000018021.

9- Brown, A. (2018). What is so special about online (as compared to offline) hate speech? Ethnicities, 18(3), 297 - 326.https://doi.org/10.1177/1468796817709846.

10- Cohen-Almagor, R. (2017). Why Confronting the Internet’s Dark Side? Philosophia, 45(3), 919-929. https://doi.org/10.1017/CBO9781316226391.

11- Vijayaraghavan, Prashanth, Hugo Larochelle, and Deb Roy. "Interpretable Multi-Modal Hate Speech Detection." .

12- A. Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, “Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An N-gram and TFIDF based Approach,” IEEE Int. Adv. Comput. Conf. 2018, 2018, [Online]. Available: http://arxiv.org/abs/1809.08651.

13- Mishra, Pushkar, et al. "Author profiling for hate speech detection." arXiv preprint arXiv:1902.06734 (2019). Silva, Leandro, et al. "Analyzing the targets of hate in online social media." arXiv preprint arXiv:1603.07709 (2016).

14- Silva, Leandro, et al. "Analyzing the targets of hate in online social media." arXiv preprint arXiv:1603.07709 (2016).

15- Brettschneider, Uwe, and Ralf Peters. "Detecting offensive statements towards foreigners in social media." Proceedings of the 50th Hawaii International Conference on System Sciences. 2017.

16- P. Fortuna and S. Nunes, “A survey on automatic detection of hate speech in text,” ACM Comput. Surv., vol. 51, no. 4, 2018, doi: 10.1145/3232676.

17- Mossie, Zewdie, and Jenq-Haur Wang. "Social network hate speech detection for Amharic language." Computer Science & Information Technology (2018): 41-55.

18- Chowdhury, Arijit Ghosh, et al. "ARHNet-Leveraging Community Interaction for Detection of Religious Hate Speech in Arabic." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. 2019.

19- F. Del Vigna, A. Cimino, F. Dell’Orletta, M. Petrocchi, and M. Tesconi, “Hate me, hate me not: Hate speech detection on Facebook,” in CEUR Workshop Proceedings, 2017, vol. 1816, pp. 86–95.

20- Xue, Jia, Junxiang Chen, and Richard Gelles. "Using data mining techniques to examine domestic violence topics on Twitter." Violence and gender 6.2 (2019): 105-114.

21- Mouhssine, Errais, and Chougdali Khalid. "Social big data mining framework for extremist content detection in social networks." 2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT). IEEE, 2018.

22- Varma, Rishabh, and Sartaj Ahmad. "Mass Violence Detection Using Data Mining Techniques." World Scientific News 113 (2018): 227-234.

23- D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, and A. Vakali, “Mean birds: Detecting aggression and bullying on Twitter,” in Proceedings of the 2017 ACM Web Science Conference, 2017, pp. 13–22, doi: 10.1145/3091478.3091487.

24- Guellil, Imane, et al. "Detecting hate speech against politicians in Arabic community on social media." International Journal of Web Information Systems (2020).

25- Alfina, Ika, et al. "Hate speech detection in the Indonesian language: A dataset and preliminary study." 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS). IEEE, 2017.

26- Jaki, Sylvia, and Tom De Smedt. "Right-wing German hate speech on Twitter: Analysis and automatic detection." *arXiv preprint arXiv:1910.07518* (2019).

27- Zhang, Ziqi, David Robinson, and Jonathan Tepper. "Detecting hate speech on twitter using a convolution-gru based deep neural network." European semantic web conference. Springer, Cham, 2018.

28- Pitsilis, Georgios K., Heri Ramampiaro, and Helge Langseth. "Effective hate-speech detection in Twitter data using recurrent neural networks." Applied Intelligence 48.12 (2018): 4730-4742.

29- Waseem, Zeerak, and Dirk Hovy. "Hateful symbols or hateful people? predictive features for hate speech detection on twitter." Proceedings of the NAACL student research workshop. 2016.

30- Warner, William, and Julia Hirschberg. "Detecting hate speech on the world wide web." Proceedings of the second workshop on language in social media. 2012.

31- Yasseri, T., and B. Vidgen. "Detecting weak and strong Islamophobic hate speech on social media." Journal of Information Technology and Politics 17.1 (2019).

32- Oriola, Oluwafemi, and Eduan Kotzé. "Evaluating Machine Learning Techniques for Detecting Offensive and Hate Speech in South African Tweets." IEEE Access 8 (2020): 21496-21509.

33- Sabat, Benet Oriol, Cristian Canton Ferrer, and Xavier Giro-i-Nieto. "Hate Speech in Pixels: Detection of Offensive Memes towards Automatic Moderation." arXiv preprint arXiv:1910.02334 (2019).

34- Tang, Lei, and Huan Liu. "Leveraging social media networks for classification." Data Mining and Knowledge Discovery 23.3 (2011): 447-478.

35- Mathew, Binny, et al. "Spread of hate speech in online social media." Proceedings of the 10th ACM conference on web science. 2019.

36- K. Peppler, T. S. Encyclopedia, and T. Oaks, “Final pre-publication draft,” 2017.

37- Jensen, K. B., & Helles, R. (2011). The Internet as a cultural forum: Implications for research. New Media & Society, 13, 517-533. https://doi.org/10.1177/1461444810373531

38- B. Jeyapragash and J. I. Arputharaj, “Social Networking Tools for Research Scholars: an overview,” Indian J. Sci. Indian J. Sci. Anal. Int. Wkly. J. Sci., vol. 21, no. 74, pp. 560–564, 2015.

39- Sint, Rolf, et al. "Combining unstructured, fully structured and semi-structured information in semantic wikis." CEUR Workshop Proceedings. Vol. 464. 2009.

40- Praveen, Shagufta, and Umesh Chandra. "Influence of structured, semi-structured, unstructured data on various data models." Int. J. Sci. Eng. Res 8 (2017): 67-69.

41- Sapountzi, Androniki, and Kostas E. Psannis. "Social networking data analysis tools & challenges." Future Generation Computer Systems 86 (2018): 893-913.

42- Tavassoli, Sude, Markus Moessner, and Katharina Anna Zweig. "Constructing social networks from semi-structured chat-log data." 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014). IEEE, 2014.

43- Watanabe, Hajime, Mondher Bouazizi, and Tomoaki Ohtsuki. "Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection." IEEE access 6 (2018): 13825-13835.

44- Davidson, Thomas, et al. "Automated hate speech detection and the problem of offensive language." arXiv preprint arXiv:1703.04009 (2017).

45- MacAvaney, Sean, et al. "Hate speech detection: Challenges and solutions." PloS one 14.8 (2019): e0221152.

46- Kumar, Shubham, et al. "Twitter Sentiment Analysis: A Novel Machine Learning Approach." International Journal of Control and Automation 13.4 (2020): 412-421.

47- Adedoyin-Olowe, Mariam, Mohamed Medhat Gaber, and Frederic Stahl. "A survey of data mining techniques for social media analysis." arXiv preprint arXiv:1312.4617 (2013).

48- Han, Jiawei, Micheline Kamber, and Jian Pei. "Data mining concepts and techniques third edition." The Morgan Kaufmann Series in Data Management Systems (2011): 83-124.

49- Bharati, Mrs, and M. Ramageri. "Data mining techniques and applications." (2010).

50- Zaki, Mohammed J., and Limsoon Wong. "Data mining techniques." *Selected Topics in Post-Genome Knowledge Discovery*. 2004. 125-163.

51- Gupta, Vibhuti, and Rattikorn Hewett. "Adaptive Normalization in Streaming Data." Proceedings of the 2019 3rd International Conference on Big Data Research. 2019.

52- Shannon, Claude Elwood. "A mathematical theory of communication." ACM SIGMOBILE mobile computing and communications review 5.1 (2001): 3-55.

53- Rajaraman, Anand, and Jeffrey David Ullman. Mining of massive datasets. Cambridge University Press, 2011.

54- Wu, Wei, et al. "A review for weighted minhash algorithms." IEEE Transactions on Knowledge and Data Engineering (2020).

55- Broder, Andrei Z. "On the resemblance and containment of documents." Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171). IEEE, 1997.

56- F. Ahmed Sagar, “Cryptographic Hashing Functions-MD5,” no. September, pp. 1–9, 2016.

57- Tiwari, Harshvardhan, and Dr Asawa. "A secure hash function MD-192 with modified message expansion." arXiv preprint arXiv:1003.1492 (2010).

58- Habeeb, Ahmed. "Secure Hash Algorithm."

59- McDaniel, Bill. "An algorithm for error correcting cyclic redundance checks." CC PLUS PLUS USERS JOURNAL 21.6 (2003): 6-17.

60- Aizawa, Akiko. "An information-theoretic perspective of tf–idf measures." Information Processing & Management 39.1 (2003): 45-65.

61- Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." International conference on machine learning. 2014.

62- Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.

63- Wang, Lidong, and Cheryl Ann Alexander. "Machine learning in big data." International Journal of Mathematical, Engineering and Management Sciences 1.2 (2016): 52-61.

64- M. Steinbach, G. Karypis, and V. Kumar, “A Comparison of Document Clustering Techniques,” KDD Work. text Min., vol. 400, pp. 1–2, 2000.

65- N. S. Chauhan, “Decision Tree Algorithm — Explained,” Towards Data Science, 2019. [Online]. Available: https://towardsdatascience.com/decision-tree-algorithm-explained-83beb6e78ef4. [Accessed: 27-Mar-2020].

66- D. K, “Top 5 advantages and disadvantages of Decision Tree Algorithm,” medium, 2019. [Online]. Available: https://medium.com/@dhiraj8899/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a. [Accessed: 27-Mar-2020].

67- R. Pupale, “Support Vector Machines (SVM) — An Overview,” Towards Data Sciencedatascience, 2018. [Online]. Available: https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989. [Accessed: 08-Mar-2020].

68- D. K, “top 4 advantages and disadvantages of support vector machine,” medium, 2019. [Online]. Available: https://medium.com/@dhiraj8899/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107. [Accessed: 14-Mar-2020].

69- W. Jamal, S. Das, I.-A. Oprescu, K. Maharatna, F. Apicella, and F. Sicca, “Classification of autism spectrum disorder using supervised learning of brain connectivity measures extracted.

70- Goldberg, Yoav. "A primer on neural network models for natural language processing." Journal of Artificial Intelligence Research 57 (2016): 345-420.

71- J. Brownlee, “machine learning mastery.” [Online]. Available: https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data.