University of Kerbala

College of Computer Science & Information Technology

Computer Science Department

CYBERBULLYING CLASSIFICATION AND DETECTION IN TWITTER USING DATA MINING TECHNIQUES

A Thesis

Submitted to the Council of the College of Computer Science & Information Technology / University of Kerbala in Partial Fulfillment of the Requirements for the Master Degree in Computer Science

Written by

Fatima Nadi_Ali Hussein

Supervised by

Asst. Prof. Dr. Hiba Jabbar Aleqabie

2024 A.D. 1445 A.H

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ﴿١﴾
خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ ﴿٢﴾
اقْرَأْ وَرَبُّكَ الْأَكْرَمُ ﴿٣﴾
الَّذِي عَلَّمَ بِالْقَلَمِ ﴿٤﴾
عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ ﴿٥﴾

صدق الله العظيم

Supervisor Certification

I certify that the thesis entitled (**Cyberbullying Classification and Detection in Twitter Using Data Mining Techniques**) was prepared under my supervision at the department of Computer Science/College of Computer Science & Information Technology/ University of Kerbala as partial fulfillment of the requirements of the degree of Master in Computer Science.

Signature:

Supervisor Name: Asst. Prof. Dr. Hiba Jabbar Aleqabie

Date: 16 / 3 /2024

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "(**Cyberbullying Classification and Detection in Twitter Using Data Mining Techniques**" for debate by the examination committee.

Signature:

Assist. Prof. Dr. Muhannad Kamil Abdulhameed

Head of Computer Science Department

Date: / /2024

Certification of the Examination Committee

We hereby certify that we have studied the dissertation entitled (**Cyberbullying Classification and Detection in Twitter Using Data Mining Techniques**) presented by the student (**Fatima Nadi_Ali Hussein**) and examined her in its content and what is related to it, and that, in our opinion, it is adequate with (**Excellent**) standing as a thesis for the Master degree in Computer Science.

Signature:	Signature:
Name: Noor Dhia Al-Shakarchy	Name: Mohsin Hasan Hussein
Title: Assist. Prof. Dr	Title: Assist. Prof. Dr
Date: / /2024	Date: / /2024
(Chairman)	**(Member)**
Signature:	Signature:
Name: Ehsan Ali Kareem Al-Zubaidi	Name: Hiba Jabbar Aleqabie
Title: Assist. Prof. Dr	Title: Assist. Prof. Dr
Date: / /2024	Date: / /2024
(Member)	**(Member and Supervisor)**

Approved by the Dean of the College of Computer Science & Information Technology, University of Kerbala.

Signature: Mowafak

Asst. Prof. Dr. Mowafak Khadom Mohsen

Date: / /2024

(Dean of College of Computer Science & Information Technology)

Dedication

Dedicated with love to...

My dear family, whose love and support made all of this achievable.

In loving memory of my father.

My mother, who has consistently cared for me with her affection and dedication.

My wonderful husband, Muhammed Awad, who has been by my side through thick and thin.

My children, Lana and Adam.

And my sister, Zainab, who has been like a second mother to me, standing by and offering unwavering support.

Fatima N. Ali

Acknowledgement

I want to express my gratitude for the success I've achieved, knowing it's by God's will and grace. I thank God for His support throughout my research journey.

I'm thankful for my professors in the Department of Computer Science and Information Technology, especially my supervisor, Dr. Hiba showed great commitment, closely monitoring my progress, offering valuable feedback, and being patient during challenging times.

I'm deeply grateful to my late father. To my dear mother, your love and encouragement mean the world to me.

To my loving husband Muhammed Awad for his constant support my devoted husband, your believe in me has been crucial to my success.

To my beloved children, Lana and Adam, you light up my life.

I'd like to thank my brothers (Hussein, Abbas, Ahsan, Emad, and Ayad) and my sister, Zainab. Your continued support is invaluable.

Thankful to My father-in-law Dr. Awad Khadim, for his constant encouragement and guidance. His wisdom and advice have been invaluable in motivating me to strive for success.

My mother-in-law, you have always played a great role in encouraging me and pushing me toward achieving my ambitions. Thank you for your friendship and unconditional love.

My sisters-in-law (Alaa, Esraa, Arwa, Zainab and Almaha) thank you for supporting and stand with me

My appreciative friends Ghayda, Vinas, Mais, and Liqaa have always been there for me, providing strength during my weakest moments.

Abstract

The rapid growth of social media has given rise to new forms of bullying. Facebook, Twitter, and YouTube platforms have become a significant concern for individuals, organizations, and society as a whole. The early detection and intervention of cyberbullying on social media are critical to mitigating its harmful effects.

The proposed system involved two models. The first model included two multi-classification datasets and worked with text mining to classify the tweets into multi classes using different techniques. The second model utilized social network analysis (SNA) to detect the influential users that disseminated the bullying in communities and the bullying content associated with it.

In the first model, several techniques used in the feature extraction step are TF-IDF with Bow and Word2Vec. For the classification, four supervised machine learning algorithms, Random Forest (RF), Support Vector Machine (SVM), K-nearest neighbors (KNN), and Naïve Bayes (NB), are utilized. The second model used three centrality measures: degree centrality (DC), betweenness centrality (BC), and closeness centrality (CC).

The results of the first model demonstrated the effectiveness of the first dataset, the "Cyberbullying Classification Dataset," with result accuracy and precision rates of 93% and 87%, respectively, with minimal computational time. While the second dataset, “Cyberbullying Types Dataset,” got results with accuracy and precision rates of 89% and 90%, respectively, these results led us to select the "Cyberbullying Classification Dataset" as a suitable candidate for Social Network Analysis (SNA).

In conclusion, SNA revealed valuable insights into cyberbullying detection, with a particular focus on frequent user mentions (influential users) and high centrality measures as reliable indicators. The stability of hashtags over time also played a critical role in identifying problematic content.

Declaration Associated with this Thesis

Some of the works presented in this thesis have been accepted as listed below.

1. F. N. A. Hussein and H. J. Aleqabie, “Cyberbullying Detection on Social Media: A Brief Survey,” published in *the Second International Conference on Advanced Computer Applications*, 2023.
2. F. N. A. Hussein and H. J. Aleqabie, “Cyberbullying Detection in Twitter Conduction Graph Mining and Machine Learning,” published to the journal of Kerbala University, 2023.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Description
BC	Betweenness Centrality
BOW	Bag of Word
CBCD	Cyberbullying classification dataset
CBTD	Cyber Bullying Types dataset
CC	Closeness Centrality
DC	Degree Centrality
DT	Decision Tree
KNN	K-Nearest Neighbours
ML	Machine Learning
MNB	Multinomial Naïve Bayes
RF	Random Forest
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
NB	Naïve Bayes

CHAPTER ONE

INTRODUCTION

1.1 General Overview

In the present era of advanced technology, the internet is predominantly utilized for communication, a trend that may inadvertently foster harmful behaviors. One prominent example of such disruptive and detrimental conduct is cyberbullying. Research suggests cyberbullying represents a shift from traditional offline bullying to online tactics, primarily executed through social media platforms [1]. Cyberbullying is a deliberate and aggressive act perpetrated by an individual or a group against a victim who lacks immediate means of self-defense, utilizing persistent electronic, digital, and multi-modal communication and interaction [2]. It constitutes a relatively new form of bullying that significantly differ from conventional harassment. Unlike traditional bullying, cyberbullying is not confined by time or location constraints, and its potential for harm is amplified by the anonymity it affords, potentially reaching a wider audience and resulting in more severe abuse. The prevalence of cyberbullying has surged, particularly among younger generations, owing to the widespread accessibility of the internet and the popularity of social media platforms such as Facebook, Instagram, and Twitter. This aggressive behavior profoundly affects victims' mental well-being and lives and can serve as a model for imitation by other adolescents or group members [3].

Cyberbullying can take place through a variety of mediums, including but not limited to text messages, instant chats, social networking sites, and online games. According to data compiled by statisticbrain.com, Facebook is the most popular social media site for cyberbullying [4]. The following are the most typical and common media where cyberbullying can happen:

1. **Electronic mail (email):** a method of exchanging digital messages from an author to one or more recipients.
2. **Instant messaging:** a type of online chat that offers real-time text transmission between two parties.
3. **Chat rooms:** a real-time online interaction with strangers with a shared interest or other similar connection.
4. **Text messaging:** the act of composing and sending a brief electronic message between two or more mobile phones.
5. **Social networking:** a platform to build social networks or social relations among people who share interests, activities, backgrounds or real-life connections.
6. **Websites:** a platform that provides service for personal, commercial, or government purpose.

There are several types of cyberbullying that distinguished in [5][6]:

1. **Flooding**: involves the bullies sending repeated frequent nonsensical comments/posts in order tonot allow the targeted victim to participate in the conversation.
2. **Masquerade:** involves the bullies pretending to mimic or impersonate the target victim.
3. **Flaming and bashing** involves an online fight where the bully sends and/or posts insulting, hurtful and vulgar contents to the targeted victim privately or publicly in an online group.
4. **Trolling** involves purposely publishing comments which disagree with other comments in order to incite arguments or negative emotions

although the comments themselves might not be vulgar or hurtful in themselves.

5. **Harassment** is the kind of conversation where the bullies frequently send insulting and rude messages to the victim privately.
6. **Denigration** occurs when the bullies send or publish gossips or untrue statements about the victims to damage the victims' friendships/reputations.
7. **Outing** occurs when bullies send or publish private or embarrassing information in public chat-rooms or forums. This type of cyberbullying is similar to denigration. However, in the outing, there might be a relationship between bully and victim
8. **Exclusion** involves intentionally excluding someone from an online group. This type of cyberbullying happens among youth and teenagers more prominently.
9. **Impersonation** is the act of pretending to be someone else and transmitting or publishing information intended to place that person in danger or problems, or to harm their reputation or friendships.
10. **Trickery**: persuading a person to divulge secrets or humiliating information in order to publish it online.
11. **Cyberstalking** is characterized by repeated, persistent harassment and denigration that involves threats or induces substantial anxiety [5], [6].

1.2 Problem Statement

1. Locate the multiclassification dataset from the IEEE.

2. Traditional machine learning methods are ineffective in identifying influential users due to a lack of relevant user information in datasets.
3. the identification of nuanced types of cyberbullying that conventional text analysis tools would find difficult to identify.

1.3 Challenge

1. Data ambiguity and noise: Cyberbullying cases are hard to identify and categorize due to Twitter data's frequent noise and ambiguity. False positives and negatives as well as incorrect tweet classification may result from this.
2. Contextual understanding: Since particular words or phrases may be used differently in different contexts, it is essential to comprehend the context of tweets in order to accurately detect cyberbullying.

1.4 Objectives

1. Build a model for cyberbullying classification (CBC) using several ML algorithms.
2. Build a model for cyberbullying detection (CBD) to detect influential node (user).

1.5 Related Works

This section provides a concise literature review of previous research on detecting cyberbullying in social networks. The primary aim is to present an overview of prior studies, highlighting the challenges addressed in this thesis and providing a summary of the most pertinent works within the scope of the literature review.

The authors K. Das, S. Samanta, and M. Pal in 2018 [7] took a supervised approach to detect cyberbullying. They extracted features using various machine learning classifiers, TFIDF, and sentiment analysis algorithms. They assessed the classifications using different n-gram language models. Their findings showed that a neural network using 3-grams achieved a higher accuracy of 92.8% compared to SVM with 4-grams, which achieved 90.3%. Moreover, the neural network outperformed other classifiers on the same dataset in another study. The dataset, sourced from Kaggle (Formspring. me), comprises 1608 instances of English conversations, categorized into two classes: Cyberbullying and non-cyberbullying, each containing 804 instances.

Balakrishnan et al. 2019 have identified patterns of bullying among Twitter groups by examining the associations between personality factors and instances of cyberbullying. The RF method, recognized in machine learning, was used to classify cyberbullying into several categories, such as aggressor, spammer, bully, and normal. They developed an automatic cyberbullying detection mechanism based on Twitter users' psychological characteristics, personalities, and sentiments This classification was carried out with a baseline algorithm that included several Twitter variables, including the number of mentions, number of followers and following, and popularity [8].

In a study by Nurek in 2020 [9], the assessment focused on integrating social network measures with additional features obtained through feature engineering for classifying members within an organizational social network. Machine learning techniques were applied for this classification task, involving the evaluation of Decision Trees, Random Forest, Neural Networks, and Support Vector Machines. Furthermore, the study introduced a collective classification algorithm. This approach enabled a comparison between the performance of conventional machine learning classification methods, enhanced by social network analysis, and a conventional graph algorithm typically used in such scenarios.

Choi 2021 focuses on a substantial alternative to blocking harmful comments by identifying prominent offenders using text mining and social network analysis (SNA). They chose the Korean online community Daum Agora based on postings and comments via web crawling. They compute the Losada ratio, which is a positive-to-negative comment ratio. Then, using text mining, propose and construct a cyberbullying index. They employ the SNA approach to analyze user interactions in order to determine the effect that the core users have on the community. Through real-world applications and assessments, they verify the suggested approach to identifying essential cyberbullies. The suggested approach has implications for online community management and minimizing cyberbullying [10].

Wang 2021 segmented tweets by considering social network connections, compiled all the nodes involved and constructed a graph using retweet relationship and follower relationships. They acquired datasets from Twitter and conducted experiments using various training models, including RF, SVM, LR, AdaBoosting, Parsimonious Bayes, SGD, CNN, and LSTM [11].

Mahmud 2022 in [12], the author employed the Cyberbullying Classification Dataset, one of the datasets utilized in this thesis. The study encompassed the application of five distinct machine learning models, LightGBM, XGBoost, LR, RF, and AdaBoost, to identify instances of cyberbullying using textual features as input. LightGBM demonstrated remarkable performance, outperforming the other models and achieving noteworthy results, including an accuracy rate of 85.5%, a precision rate of 84%, a recall rate of 85%, and an F-1 score of 84.49%.

In a study conducted by Ioannis in 2023 [13], four datasets were collected from sources, including IEEE, Zenodo, and Kaggle, for cyberbullying detection. Two datasets, namely the Cyber Bullying Types Dataset IEEE and the Cyberbullying Classification Dataset, were utilized in this thesis. Ioannis examined these datasets individually and in combination, employing eight classification algorithms and (NLP) and (ML) techniques to determine the most effective model for cyberbullying detection. Ioannis experimented with a range of classification algorithms for cyberbullying detection, including LR, DT, RF, XGBoost, Multinomial NB, SVM, Bagging DT, and Boosting DT. In the Cyber Bullying Types Dataset IEEE, LR achieved an accuracy of 91% and a precision of 99% when using the TF-IDF technique. In the Cyberbullying Classification Dataset, SVM with TF-IDF/CountVectorizer achieved a detection accuracy of 85%. In the table 2.2 illustrates the Summary of Literature Review.

Table 1.1: Summary of Literature Review

N	Ref.	Dataset	Techniques	Evaluation metric	SNA F.
1	[7]	Binary	SVM TFIDF	Acc=92.8%	
2	[8]	Binary	RF	Recall=95%	number of mentions, number of

3	[9]	Binary	rule-based	Recall=95%	followers and following and popularity, Facebook comments
4	[10]	Daum Agora data	Losada ratio, Centrality	-	Posts, Comments
5	[11]	Binary	SVM (linear)	Acc=91%	Retweets, followers
6	[12]	Binary	LightGBM	Acc=85.5%,	-
7	[13]	Binary	LR , SVM with TF-IDF/CountVectorizer	Acc= 91% Precision 85%.	- -
8	CBD.SNG	Multi-classification	RF	Acc=93%	Users mention, hashtags

1.6 Thesis Layout

The thesis is divided into five chapters. The summaries of the chapters are as follows:

Chapter One: Introduction.

Chapter Two**:** This chapter details the theoretical background used in this thesis.

Chapter Three: This chapter concentrates on the proposed method.

Chapter Four: This chapter discussed the obtained results

Chapter Five: This chapter involves conclusions and suggestions for future works.

CHAPTER TWO

THEORETICAL BACKGROUND

2.1 Overview

In this chapter, we delve into the theoretical underpinnings that form the foundation of the methods employed throughout this thesis. Establish a robust understanding of the techniques and approaches utilized in our research. This chapter comprehensively explores the theoretical framework. We will examine the core principles, concepts, and methodologies that underlie each method, providing readers with the necessary background to comprehend the subsequent chapters' implementation and findings. By elucidating the theoretical background, we aim to bridge the gap between theory and application, enhancing the clarity and depth of our research.

2.2 Cyberbullying

"Cyberbullying" encompasses any bullying that occurs online or digital. It can take the form of text-based exchanges, messages, comments, forum posts, or the sharing of images, and it can happen on various devices such as smartphones, laptops, and other digital platforms. Cyberbullying involves the recurrent transmission of hurtful or offensive content by an individual or a group on social media platforms with the intent of causing harm or emotional distress to others [14].

2.3 Text preprocessing

Preprocessing procedures are necessary to enhance data suitability for data mining. Data preprocessing encompasses many strategies and techniques, and these approaches are intricately interconnected. It involves a series of steps aimed at optimizing data for practical data mining, and these steps can vary based on the specific context and requirements of the data analysis task [15].

2.3.1 Text Cleaning

Data cleansing is a process that removes inaccurate, incomplete, or irrelevant data from a dataset, focusing on removing duplicate records and data that doesn't contribute to the overall dataset quality [15].

- Remove Numbers: Eliminate numerical digits from the text.
- Remove Punctuation: Remove punctuation marks from the text.
- Remove Whitespaces: Remove extra whitespaces and ensure uniform spacing.
- Eliminate characters like [[.].@...: Remove specific characters like brackets, dots, and '@' symbols.
- Eliminate Hashtags: Remove hashtags (e.g., "#machinelearning").
- Correct Contractions: Expand contractions (e.g., "don't" to "do not").
- Handle Effect Negations: Handle negations (e.g., "not good" to "not_good").
- Lowercasing: Convert text to lowercase.

- Replace Elongated Words: Replace repeated characters to shorten elongated words (e.g., "loooove" to "love").
- Eliminate URLs: Remove URLs or website links (e.g., "https://example.com").

2.3.2 Stopword Removal

Stopword removal, as discussed in references [16]–[19] constitutes a crucial preprocessing step in text analysis. This process involves excluding common words, often referred to as stopwords, from the text, as these words typically carry little or no substantive meaning and can hinder the user's ability to extract meaningful insights from the text. Stopwords are typically compiled into a stop list [17], a reference for identifying and removing these noise words.

Additionally, this procedure can include removing other forms of textual noise, such as special characters and superfluous symbols (e.g., "the," "and," "in") that don't carry significant meaning in the context. While stopword removal aids in reducing the number of non-informative words that appear frequently, it ensures that the focus remains on the essential content of the text. [19].

2.3.3 Tokenization

The tokenization process breaks down a text into individual units or tokens, such as words or phrases, for further analysis and processing, explained in [18], [19]. It involves segmenting sentences into individual words while eliminating any punctuation marks irrelevant to the task. This essential text preprocessing step facilitates the transformation of continuous text into discrete units, allowing for more effective analysis and data processing. Subsequently, the data proceeds to the data weighting stage, where the significance and importance of the tokenized words are often assessed and quantified for various text analysis tasks. By breaking down sentences into words and removing extraneous punctuation, tokenization is a fundamental procedure that lays the foundation for more advanced text analytics and natural language processing tasks.

2.3.3 Lemmatization

Reducing the inflected words properly and ensuring that the root word belongs to the language. It's usually more sophisticated than stemming, since stemmers works on an individual word without knowledge of the context. In lemmatization, a root word is called lemma. A lemma is the canonical form, dictionary form, or citation form of a set of words. Reduce words to their base or root form (e.g., "running" to "run") [20].

2.3.4 Expanding contractions and abbreviations

Expanding contractions and abbreviations, as discussed in reference [21], serve the purpose of streamlining and enhancing text comprehension. This process involves replacing contractions, abbreviations, and acronyms with their complete, unabbreviated forms to maintain consistency throughout the text. For instance, it ensures that "IT" is unequivocally understood as "information technology" rather than being potentially confused with other interpretations. While this expansion can increase the statistical robustness of text analysis, it may also diminish the ability to capture the nuances of speech or writing style.

Moreover, it is crucial to acknowledge that errors can be introduced during the expansion process, potentially affecting the overall validity of the text. In summary, expanding contractions and abbreviations aims to promote clarity and precision but must be executed cautiously to preserve the text's integrity [21].

2.3.5 Resampling

Data imbalance in Machine Learning refers to an unequal distribution of classes within a dataset, a common issue encountered primarily in classification tasks. This problem arises when the classes or labels in a dataset are not evenly distributed. To address this challenge, a commonly employed solution involves resampling methods, either adding records to the minority class or removing records from the majority class.

A prevalent approach to tackling this issue is to employ resampling techniques, as referenced in [22], [23]. Two primary techniques are commonly employed to address class imbalance in a dataset: undersampling and oversampling. Undersampling involves reducing the number of instances in the majority class, thereby restoring balance to the dataset. Conversely, oversampling focuses on increasing the representation of the minority class by replicating data points. This approach helps create a more balanced dataset where both class groups have comparable instances. Achieving this balance is essential as it ensures that machine learning classifiers give equal importance to both classes, enhancing the model's ability to learn and generalize effectively[23].

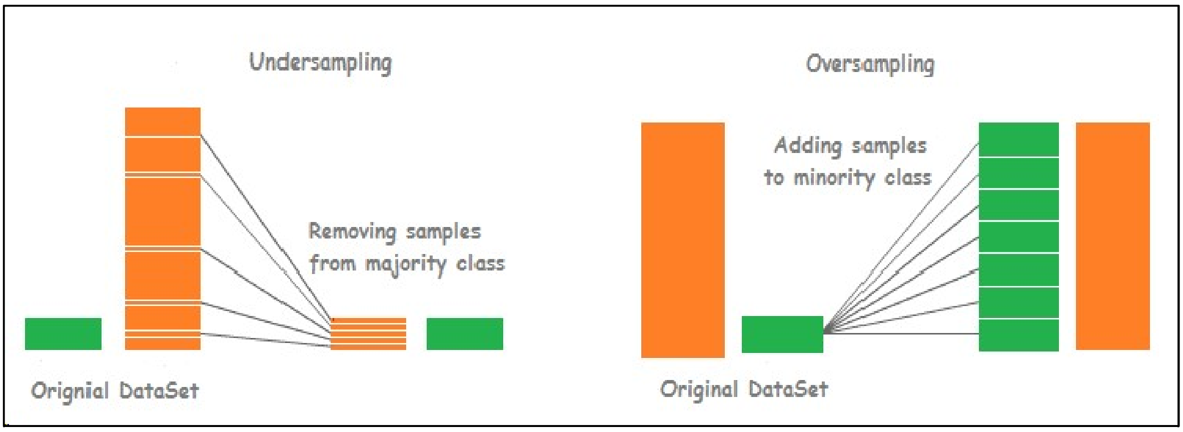

Undersampling
Removing samples from majority class
Orignial DataSet
Oversampling
Adding samples to minority class
Original DataSet

Figure 2.1:Differences Between Undersampling And Oversampling [23]

2.4 Feature extraction Techniques

Raw data must be transformed into a quantitative, computable representation. It must be amenable to representation as a fixed length vector of features, as the vast majority of ML methods require a fixed length input (of any arbitrary length). This is a completely different task for every domain: in the case of computer vision, it means extracting raw pixel values (from images or “crops” of identical sizes). In the case of stocks or Electroencephalograms, it involves extracting time-series and amplitudes. In the case of natural text, it may involve counting the words or characters (“Bag of Words”) [24]. A typical machine learning classification pipeline is shown in Figure 2.2.

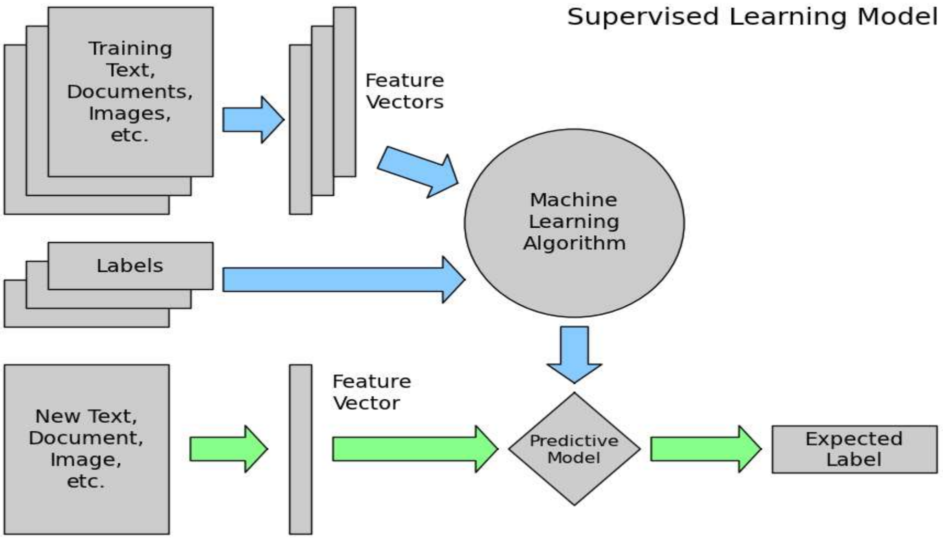

Supervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Labels
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

Figure 2.2: ML Feature Extraction [24]

2.4.1 TF-IDF

TF-IDF, which stands for Term Frequency-Inverse Document Frequency, is a statistical technique for assessing the significance of a word within a document or a broader corpus context. This method considers a word important when frequently occurring within a specific document. However, its importance diminishes as the word becomes more prevalent across the entire corpus. Suppose a word frequently appears in a particular article or document while relatively rare in other documents or articles. In that case, it becomes a more representative feature of that specific content[11]. The following equation provides a statistical representation of how a word is weighted within a text using TF-IDF [25]:

tf(t,d) = count of t in d / number of words in d (2.1)

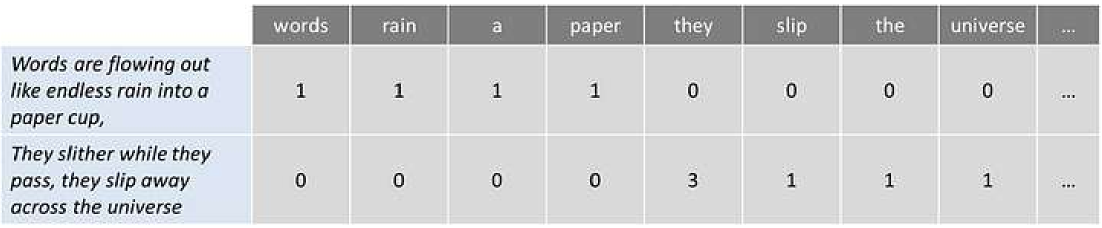
idf(t) = log(N/ df(t)) (2.2)

tf-idf(t, d) = tf(t, d) * idf(t) (2.3)

In this equation, (N) represents the total number of texts, and (t) signifies the total count of text documents that contain the word (t) in the dataset.

2.4.2 Bag-of-Words (BoW)

The Bag-of-Words (BoW) The Bag of Words model is a commonly used approach that involves counting all words in a piece of text. Essentially, it creates an occurrence matrix for a sentence or document, disregarding grammar and word order. Figure 2.3 show word frequencies or occurrences are then used as features for training a classifier [13].



	words	rain	a	paper	they	slip	the	universe	...
Words are flowing out like endless rain into a paper cup,	1	1	1	1	0	0	0	0	...
They slither while they pass, they slip away across the universe	0	0	0	0	3	1	1	1	...

Figure 2.3: Example Bag of Words [11]

2.4.3 Words Representation (Words Embeddings)

Most natural language processing applications require a word representation stage, which is a sort of learned representation that enables similar meaning words to have the same representation. Hence, many approaches to representing words as dense vectors in a low-dimensional vector space have been developed, each adopting a different training strategy inspired by neural network language modeling [26]. Word2vec and Global Vectors (GloVe) are two effective deep learning approaches for word embeddings [27]. In this thesis, we will discuss word2vec that used for learning word embeddings:

A. Word2vec

Word2vec is a word representation model created at Google in 2013 by Tomas Mikolov [28]. This model employs two hidden layers in a shallow neural network to generate a vector for each word. Word vectors could be obtained using two methods: Continuous Bag of Words (CBOW) and Skip Gram (SG) models. In order to get a better representation of words, it is recommended to train the corpus using the huge corpus. Word2Vec has been shown to be effective in a wide range of Natural Language Processing (NLP) tasks [29].

1. Continuous Bag of Words (CBOW) Model

The context for a particular target word is provided by surrounding words in the CBOW model. The word representation is built by maximizing the (log-) probability of predicting the target word given its context. The CBOW model has a simplistic neural architecture in which the nonlinear hidden layer is eliminated and the projection layer is shared by all words [30]. The model optimizes the following for a given target word wt and its context {wt-c,…,wt+1,…,wt+c}

$$\frac{1}{|v|}\sum_{t=1}^{|v|}\log[P(w_t|\,w_{t-c},\ldots,w_{t-1},w_{t+1},\ldots,w_{t+c})] \qquad (2.4)$$

Where $|v|$ donates the number of words in the corpus and c donates the (size of the dynamic context of w_t.

2. Skip-Gram (SG) Model

In contrast to the CBOW model, the skip-gram (SG) model calculates the current word using context words. As showed in figure 2.4 it has a similar architecture in that the neural network's input and output are reversed [31]. Each word vector in a corpus is trained to maximize the (log-) probability of creating neighboring words. Given a set of training words {wt-c,…,wt+c}, the model maximizes the average (log) probability of predicting the context of the current target word [30]:

$$\frac{1}{|v|}\sum_{t=1}^{|v|}\sum_{j=t-c,j\neq t}^{t+c}\log[P(w_j|w_t)] \qquad (2.5)$$

Where $|v|$ donates the number of words in the corpus and c donates the size of the dynamic context of w_t. Figure 2.4 illustrate wod2vec models work.

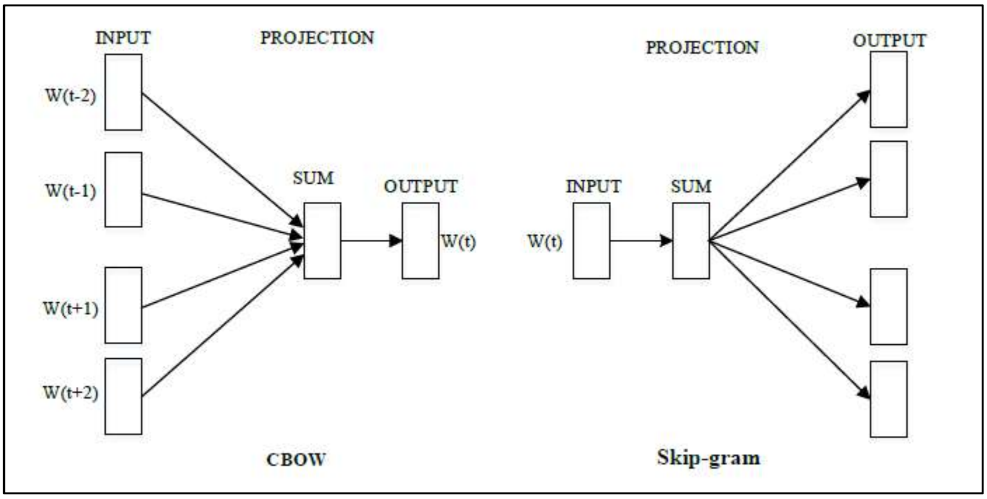


Figure 2.4: Word2vec Models [27]

B. Pre-trained Embeddings

Word Embedding is a powerful deep learning approach for creating words and documents vector representations. For training and generating an appropriate vector for each word, word2vec requires vast corpora. Google, for example, utilized big data to train word2vec algorithms and then re-released pre-trained word vectors with 300 dimensions [27]. Hence, Pre-trained Word Embeddings are embeddings learnt in one task and applied to another comparable task. These embeddings are trained on large datasets, stored, and applied to different tasks[32].

2.4.4 Cosine Similarity

Semantic similarity plays a crucial role in linguistics, particularly when determining the similarity in the meanings of words. Semantic similarity between words involves identifying similarities between two or more words. Jatnika 2019 regarding the similarity of word meanings, it is possible for two words to be

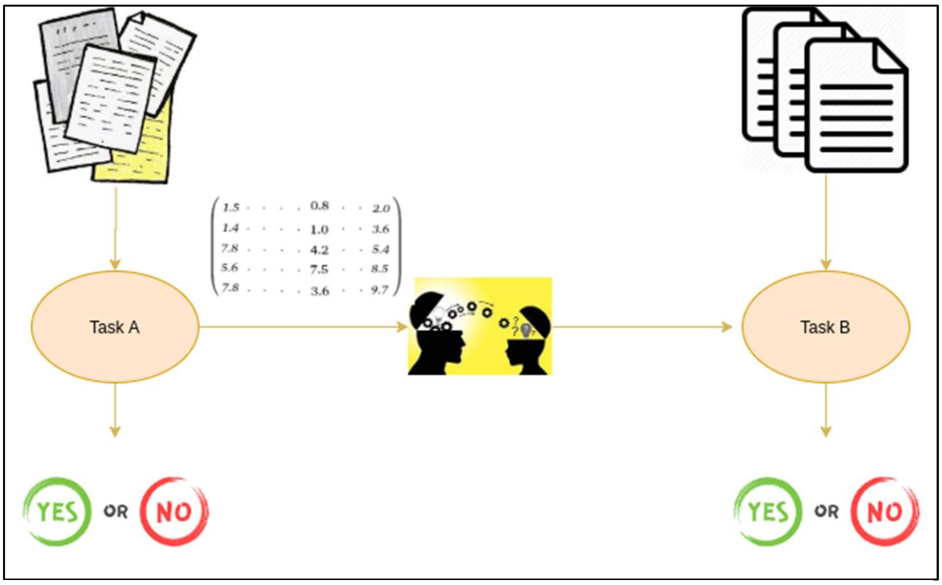


Figure 2.5: Pre-Trained Word Embeddings [32]

Different in their syntactical structure but have identical meanings. For instance, "Me" and "I" have the same meaning. The calculation of word meaning similarity has been extensively explored in linguistics, and it is often based on fundamental linguistic rules that result from human reasoning [33].

Cosine similarity is a mathematical technique employed to gauge the degree of the semantic connection between linguistic elements, concepts, or instances. It is achieved by assigning numerical values based on comparing information that characterizes their meaning or attributes [33] —for instance, understanding the likeness between a bicycle and a motorcycle or the contrast between a car and a horse. An illustration of semantic similarity is provided in table 2.1.

Table 2.1: Examples of the Word Pair Relationships By Mikolov

Relationship	**Example 1**	**Example 2**
France - Paris	Italy : Rome	Apple : Iphone

Big - Bigger	Small : Larger	Kona : Hawaii
Miami - Florida	Baltimore : Maryland	USA : Pizza
Einstein - Scientist	Messi : Midfielder	Obama : Barack
Sarkozy - France	Google : Android	Quick : Quicker

The similarity equation is defined below:

$$(x,y) = \frac{\mathrm{x.y}}{||\mathrm{x}||||\mathrm{y}||} \tag{2.7}$$

where ||x||, ||y|| are the Euclidean norm of vector x = (x1, x2,..., xp), y= (y1, y2,..., yp) respectively. A cosine value of 0 means that the two vectors have no match, while a smaller angle means the greater match between vectors [34].

2.4.5 Graph theory

In the realm of graph theory, a Graph denoted as G = (V, E) comprises two essential components: nodes and edges, serving as a means to elucidate connections within a collection of entities. These entities are referred to as Nodes and collectively form the set V. The connections themselves are denoted as edges and collectively represented by E, facilitating the linking of two nodes within the graph. When an edge connects two nodes, these nodes are termed neighbors, signifying that they share a relational association under the assumed context [35].

The Centrality is a feature that falls under the category of informative score features. The sentence's centrality implies that it is similar to other sentences. A document (or a collection of documents) is represented as a graph, with nodes representing sentences and connections connecting them weighted according to their similarity. The centrality of a node can be determined by computing its degree or by running a ranking algorithm. After calculating the centrality score for each sentence, the sentences are sorted in reverse order, with the highest-ranking ones included in the summary. If a sentence has a greater centrality degree, it is the best contender for inclusion in the summary, and its score is calculated as follows in equation 2.8 [36]:

$$(\mathbf{Si}) = \sum_{i=0}^{n} \mathrm{CosSim}(\mathrm{Si}\,, \mathrm{S}_{(\mathrm{n}-\mathrm{i})}) \qquad (2.8)$$

Where Si represents the sentence and CosSim is the mean cosine similarity distance.

Graph mining involves the process of extracting non-trivial graph structures from a single graph or a collection of graphs. It begins with feature extraction, where all text is transformed into a graph. The bag-of-words approach is a common technique used for this purpose, representing words in a text as a graph. To train the graph mining algorithm, labelled training data containing graphs derived from various text samples is used. This training process is employed to create a classification model [25].

2.4.6 Graph Network Centrality Indices

Centrality is a fundamental concept in network analysis, allowing us to pinpoint the pivotal, influential, or central nodes within a network. This concept finds application in diverse scenarios [7] [36]. For instance, within a club consisting of 100 members, the president is often regarded as central due to their leadership role. Similarly, the central headquarters takes on a pivotal role in a nationwide banking network with numerous branches. Within the confines of a classroom, the student designated as the monitor is perceived as central, while in an educational institution, such as a college, the principal is considered central among the teaching staff [7].

Centrality metrics are crucial in identifying influential elements within large datasets, especially in network-based activities like spreading viruses or disseminating information. In network analysis, identifying crucial vertices is essential. However, not all centrality measures are universally applicable and their suitability depends on the specific application. The time complexity of centrality measures is also a critical consideration. Over time, various centrality measures have been developed, catering to varying interpretations of vertex or edge importance, and applied judiciously in their respective domains. [7].

A. Degree centrality

Degree centrality is a measure of the number of nodes that are directly connected to a particular node. It is a fundamental measure of network analysis. There are two types of degree centrality: in-degree centrality and out-degree centrality. In-degree centrality counts connections that point towards a vertex, while out-degree centrality counts connections originating from a vertex and going to other vertices [36]. Mathematically, the degree centrality (C_D) of a node x is defined as the number of edges that link x to other nodes [7].

$$C_D(x) = d_x \qquad (2.9)$$

Where d_x represents the degree of node x. normalization, degree centrality can also be expressed as $C_D'(x) = d_x / (n-1)$, with n denoting the size of the network. For unweighted networks, the time complexity of this measure is O(m), where m signifies the number of edges in the network [7].

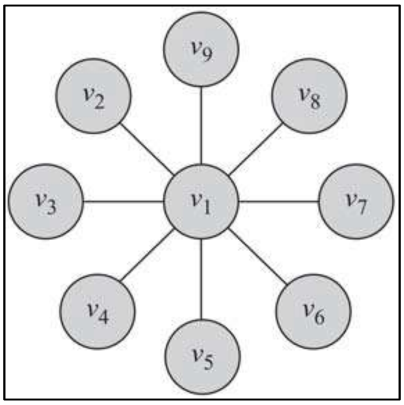


Figure 2.6 Degree Centrality Example

Figure 2.6 [37] shows a sample graph. In this graph, degree centrality for node v1 is C_D(v1) = d1 = 8, and for all others, it is C_D(vj) = dj = 1, j , 1.

A. Betweenness Centrality (BC)

The understanding of pathways plays a vital role in the examination of networks. An often-encountered inquiry in the field of network research pertains to the determination of the distance separating two persons. The measurement of this distance is determined by quantifying the least number of sequential movements between the two entities, taking into account just the connections that exist between adjacent entities. The term "geodesic distance" refers to the most direct route between two persons, and it is often used in several centrality measures [36]. BC measure is used to evaluate the importance of a node in a network by analyzing the shortest routes that go through that particular node. The statement denotes the capacity of a node to regulate the transmission of information or interactions among other nodes [38].

To calculate BC, we employ the following procedure, which can be computationally demanding for extensive networks:

Compute the shortest paths between every pair of nodes using Dijkstra's Algorithm. For each node, determine the number of shortest paths it resides on. Normalize these numbers to a range between 0 and 1.

$$C_b(v_i) = \sum_{s \neq t \neq v_i} \frac{\sigma_{st}(v_i)}{\sigma_{st}} \qquad (2.10)$$

where σst is the number of shortest paths from node s to t (also known as information pathways), and σst (vi) is the number of shortest paths from s to t that pass through vi. In other words, we are measuring how central vi's role is in connecting any pair of nodes s and t. This measure is called

betweenness centrality [37] .

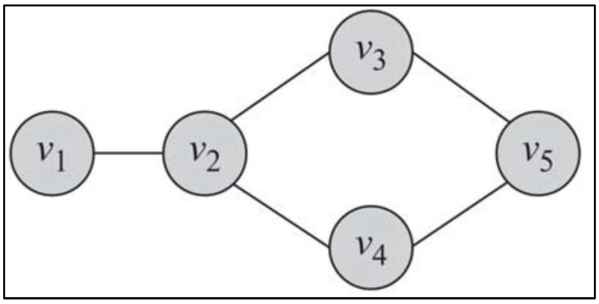


Figure 2.7 : Betweenness Centrality Example.

Dijkstra's algorithm will compute shortest paths from a single node to all other nodes. So, to compute all-pairs shortest paths, Dijkstra's algorithm needs to be run |v| - 1 times (with the exception of the node for which centrality is being computed) [37].

B. Closeness Centrality

In closeness centrality, the intuition is that the more central nodes are, the more quickly they can reach other nodes. Formally, these nodes should have a smaller average shortest path length to other nodes. Closeness centrality is defined as [37]:

$$C_c(v_i) = \frac{1}{\bar{l}_{vt}} \qquad (2.11)$$

where $\bar{l}_{vi} = \frac{1}{n-1} \sum_{vj \neq vi} l_{i,j}$ is node vi's average shortest path length to other nodes. The smaller the average shortest path length, the higher the centrality for the node.

Example: For nodes in Figure 2.7, the closeness centralities are as follows:

$$C_C\,(v1) = 1 / ((1 + 2 + 2 + 3) /4) = 0.5$$

$$C_C\,(v2) = 1 / ((1 + 1 + 1 + 2) / 4) = 0.8$$

$$C_C(v3) = C_b(v4) = 1 / ((1 + 1 + 2 + 2) / 4) = 0.66$$

$$C_C\,(v5) = 1 / ((1 + 1 + 2 + 3) / 4) = 0.57$$

Hence, node v2 has the highest closeness centrality.

The centrality measures discussed thus far have different views on what a central node is. Thus, a central node for one measure may be deemed unimportant by other measures [37].

2.5 Feature Selection

The choice of data features for training machine learning models significantly affects their performance. Features that are irrelevant or only partially relevant can harm the model's effectiveness. Feature selection is a procedure that automatically identifies the data features that have the most impact on predicting the target variable or output [39].

The advantages of conducting feature selection before modeling the data include:

- Mitigating Overfitting: It helps reduce overfitting, where the model fits the training data too closely, potentially leading to poor generalization of new data.
- Enhancing Model Performance: By eliminating less relevant data, feature selection improves modeling performance, as the model focuses on the most essential information.
- Reducing Training Time and Memory Usage: The process results in a smaller dataset, leading to faster model training and a reduced memory footprint, which can be especially beneficial for resource-intensive tasks [39].

2.5.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA), also known as the Karhunen-Loeve (K-L) method, is a supervised dimensionality reduction technique that creates a new set of variables to capture essential data information instead of selecting a subset of attributes. Initially, data is normalized to prevent attributes with large domains from dominating the analysis [40].

PCA computes k orthonormal vectors called principal components, which form a basis for the normalized data, being linear combinations of original attributes. These principal components are then sorted by significance, with the first capturing the most variance, the second the next highest, and so forth.

They essentially create new axes, unveiling data patterns and relationships. Dimensionality reduction can be achieved by keeping the top principal components simplifying data analysis and visualization while preserving vital information [40].

Mathematically, PCA involves finding eigenvalues and eigenvectors of the covariance matrix of the normalized data. Eigenvectors are the principal components, and eigenvalues signify their importance. PCA is a potent technique in data analysis and machine learning, aiding in uncovering hidden patterns, eliminating noise, and enhancing computational efficiency.

2.6 Machine learning

Machine Learning (ML), a subset of Artificial Intelligence (AI), enables systems to learn and enhance their performance through automated processes based on prior experiences without explicit programming. This learning ability is achieved by utilizing training datasets, allowing the system to make decisions [41] autonomously. This is especially useful for managing complicated tasks, mainly those involving code, as is the case with cyberbullying detection [41].

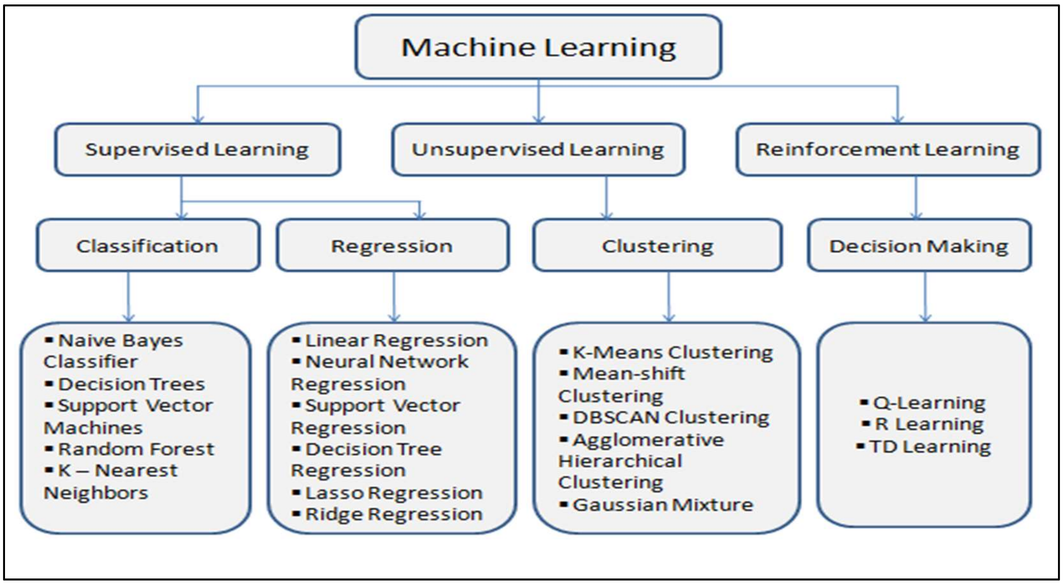


Figure 2.8: Machine Learning Types

Supervised machine learning is a subfield of ML that significantly impacts algorithmic trading by training algorithms on labelled historical data, enabling predictions or classifications on new, unseen data [42].

Unsupervised learning algorithms use unlabeled training datasets without predefined class labels to identify patterns and cluster similar data points. They focus on pattern identification and use, relying less on explicit programming [41].

Reinforcement learning is a strategy for algorithmic trading that trains agents to make sequential decisions and adjust their tactics based on market input. This system can identify the best trading strategies and adapt to changing market conditions. However, effective implementation requires careful consideration of risk management and incentive design. Further investigation is needed to fully utilize reinforcement learning in algorithmic trading [42].

Multiclass classification is a supervised ML task that uses labelled classes to predict data instance categories. The algorithm transforms textual descriptions into numeric keys, resulting in a classifier that predicts new unlabelled data instances [13].

2.6.1 The Support Vector Machine (SVM)

The Support Vector Machine (SVM) is introduced by Vladimir Vapnik and his coworkers in 1992[43] . a supervised learning model used in text classification due to its remarkable accuracy and efficiency [44]. SVM identify a hyperplane within an N-dimensional feature space, where N denotes the number of features. This hyperplane effectively segregates data points into distinct classes, making it a valuable tool in classification tasks [13].

SVMs, initially designed for binary classification, have also been used in multi-class scenarios, categorizing data into multiple classes using two-class SVMs. Multi-class classification problems involve 'K' binary classifier SVMs, where 'K' represents the number of classes. [45].

SVM provides various SVM kernel options, including linear, polynomial, Gaussian, and sigmoid kernels, and supports their use with ordinal data through two distinct approaches, as outlined below [46]:

1. One-Versus-One Classification is a method used for categorical data with multiple classes. It involves creating K(K - 1)/2 binary SVMs, each comparing a pair of classes. During prediction, a voting mechanism is used, with the highest number of predictions.

2. The one-versus-all (OvA) classification strategy is used for multi-class classification, with K classes and K binary SVMs trained to differentiate each class. Each binary classification assigns temporary labels +1 to points in the current class and -1 to points outside the class. Each K binary classifier generates a decision score f(x)k to forecast a new input x. Select the class k with the most significant f(x)k to find the final projected class ŷi. Mathematically in equation 2.11,

$$\hat{y}i = \text{argmax } [f(x)k] \text{ for } k = 1, 2, \ldots, K.. \qquad (2.11)$$

Where $f(\boldsymbol{x})_k = \sum_{j=1}^{Ns} \hat{\alpha}_{jk} y_j K_k(x_j,x) + \hat{\beta}_{0k}$. That is, for an $\boldsymbol{x}$ input, we classified the *i*th observation in the class for which $f(\boldsymbol{x})_k$ $k = 1, 2, \ldots, K$ is largest even if this evaluation is negative since this indicates that we have the highest level of confidence that the test observation belongs to the *k*th class rather than to any of the other classes. algorithm 2.1 illustrates Support Vector Machine works [47].

Algorithm 2.1 Support Vector Machine

```
Input:   Determine the various training and testing data
Output:  Predicated Class $Y$

Begin
candidate$SV$ = {closest pair from opposite classes}
while there are violating points do
        Find a violator
        candidate$SV$ = candidate$SV$ U violator
        if any $\alpha p < 0$ due to addition of c to $S$ then
           candidate$SV$ = candidate$SV$ \p
           repeat till all such points are pruneda
```

```
    end if
End
```

2.6.2 The Random Forest (RF)

The Random Forest (RF) classifier introduced by Leo Breiman in 1996 [48] , known for its ensemble approach [17], [49], is a robust machine learning algorithm employed in various applications, including cyberbullying prediction. RF mitigates overfitting issues frequently encountered in decision trees [17]. This classifier operates by constructing multiple decision-tree classifiers on diverse data subsamples and employs the average data to enhance predictive accuracy and fitting control [49]. An ensemble algorithm leverages a collection of tree models built from the training data to make predictions [17]. In multi-class classification, RF extends its capability to predict across multiple classes. However, the fundamental principle behind RF's classification mechanism remains rooted in its ability to perform majority voting among its constituent decision trees, yielding a final class prediction [49]. The following algorithm 2.2 explain the working RF [47].

Algorithm 2.2 Random Forest

Input: Training set S, features F, Class Y, number of trees B, the Weight H.

Output: Predictor of learned Tree F

```
Begin
Function RandomForest (S,F)
        H=0
            For i = 0 to B
                si = sample subset of S
            hi = RandomizedTreeLearn (si,F)
            H=H+hi
          End for
      Return H
End function.
Function RandomaizedTreeLearned (S. F)
      At each Node
            f= small subset of F
            spilt on best feature of f
            return learned tree classifier
End function
End
```

2.6.3 K-Nearest Neighbours (KNN)

K-Nearest Neighbours (KNN) is created by Thomas Cover and Peter Hart in 1967 [42]. KNN is a text classification approach that identifies the k most similar labeled instances and assigns the most common category to the unlabelled instance [16]. This non-parametric method is efficient, primarily relying on calculating distances between data points. However, its performance hinges on the choice of distance function, necessitating different functions or approximations for handling large datasets, where KNN's performance may deteriorate [16]. Additionally, the effectiveness of the KNN algorithm diminishes as the feature space's dimensions increase, known as the "curse of dimensionality" [18]. KNN work illustrate in the 2.3 algorithm [47].

Algorithm 2.3 K-nearest neighbours (KNN)

```
Input:   Dataset, evaluation settings
Output:  KNN graph predictor of K

Step 1: Load the data
    • Initialize the value of k
    • For getting the predicted class, iterate from 1 to total number of
      training data points
Step 2: Calculate the distance between test data and each row of training data.
    A. Sort the calculated distances in ascending order based on distance
       values
    B. Get top k rows from the sorted array
    C. Get the most frequent class of these rows
    D. Return the predicted class
End
```

KNN is a widely used non-parametric supervised classification algorithm known for its simplicity and efficiency. It measures similarity using the Euclidean distance and normalizes attribute values to prevent bias from attributes with varying ranges [50]. In KNN classification, an unknown pattern is assigned the most frequent class among its nearest neighbors. In case of a tie, the class with the minimum average distance to the unknown pattern is selected. A global distance function can be calculated by combining several local distance functions based on individual attributes [50].

2.6.4 Multinomial Naive Bayes (Multinomial NB)

Multinomial Naive Bayes (Multinomial NB) is widely utilized in document and text classification, particularly in the cyberbullying detection domain, as mentioned in reference [49]. This classification method is rooted in Bayes' theorem and relies on solid independence assumptions among features. It assumes a parametric model for text generation and utilizes training data to estimate optimal model parameters. Multinomial NB is suitable for handling continuous and categorical features as distinct functions, simplifying the estimation of high-dimensional density to one-dimensional kernel density estimation. In multi-class classification, the Naive Bayes algorithm can be represented by the following equation (2.12):

$$P(C_k|x)= \frac{P(x|\ Ck\ P(Ck))}{P(x)} \qquad (2.12)$$

Where:

- $P(C_k\ |x)$ is the posterior probability of class C_k given input x.
- P(x| Ck) is the likelihood of observing input x given class Ck.
- P(Ck) is the prior probability of class Ck
- P(x) is the marginal probability of observing input x

As shown in the algorithm of Naive Bayes [47].

Algorithm 2.4   Naive Bayes

```
Input:     Training/testing dataset T, F= (fl, f2, f3.., fn)
Output:    Estimated class K

Begin
Step 1: Read the training dataset T.
Step 2: Calculate the mean and standard deviation of the predictor variables in each class.
Step 3: Repeat Calculate the probability of fi using the gauss density equation in each class;
    Until the probability of all predictor variables (fl, f2, f3,., fn) has been calculated.
Step 4: Calculate the likelihood for each class.
Step 5: Get the greatest likelihood;
End
```

2.7 Social Network Analysis

Social Network Analysis (SNA) is a crucial tool for identifying and addressing cyberbullying on social media platforms like Twitter. It examines relationships and interactions among individuals, providing insights into their behavior and impact. By analyzing communication patterns such as retweets, mentions, and replies, SNA can identify key participants in cyberbullying incidents [37]. For example, [44] utilized SNA techniques to identify influential users disseminating hate speech and cyberbullying on the platform. Similarly, [48] employed SNA to identify cyberbullying by analysing user interactions and recognizing groups of aggressors and victims.

Social networks are complex data analysis tools that use graphs to represent entities or items, with nodes representing entities and edges representing relationships with varying degrees of association.[51], [52].

Social graphs can take on different forms. Like the Facebook friends graph, they are frequently undirected, where the relationships lack a specific direction. However, in scenarios like Twitter or Google+, social graphs manifest as directed graphs, where edges indicate a one-way relationship, such as followership [52].

SNA is a study that uses various methodologies to understand the intricate web of human relationships, including familial ties, friendships, organization affiliations, and social media participation. It consists of finite groups of actors, defining the network's essence through their interplay [9]. SNA offers a systematic method to analyze social networks and identify cyberbullying patterns, revealing key actors, influential users, and prevalent communities or clusters through the analysis of connections and interactions. [9].

Researchers use SNA to detect cyberbullying on Twitter by analyzing user mentions and replies, constructing network graphs to identify frequent bullies or bullying-prone users. [52]. Sentiment analysis techniques can identify negative tweets within a network graph, potentially indicating cyberbullying, by determining the presence of positive, negative, or neutral sentiment. [52].

SNA can identify influential users in a network who spread harmful content or encourage bullying. By using centrality measures, interventions can be targeted to mitigate cyberbullying. Recent studies show SNA can detect cyberbullying on Twitter using ML algorithms for accurate classification. [36].

2.8 Twitter Terminology

The Twitter data model and its fundamental terminology revolve around a "tweet," a short message limited to 280 characters (previously 140 characters until November 2018). A tweet can encompass text, images, videos, and URLs. Additionally, tweets may feature hashtags and user mentions, vital elements of Twitter communication [53].

2.8.1 Hashtags

Hashtags, identified by the "#" symbol before a word (e.g., "#funny"), are distinct and searchable keywords in tweets. They have evolved into a significant social phenomenon, with widespread use in both online and offline media. Hashtags succinctly symbolize and represent a single word or phrase in a brief message. This practice, known as "social tagging," plays a pivotal role in Twitter and microblogging in general. Metrics used to assess hashtags encompass frequency (how often a hashtag is used), specificity (its relevance to the context), consistency (its presence across different communities), and stability (how well it maintains its frequency and thematic content over time).

2.8.2 Trends

Popular hashtags and standard search terms are presented as "trends." These trends can vary by geographic region, and the topics displayed to users depend on their location and the interests of the users they follow. Analyzing Twitter's trends provides valuable insights into real-world events' significance, duration, and impact. Twitter is often viewed as a content aggregator, influencing specific trends and driving them to popularity.

2.8.3 Retweets, Mentions, Replies, and URLs

Users can "retweet" or repost tweets from other users. Additionally, users can explicitly reference another user by using a "mention" in a tweet, denoted by the "@" symbol followed by a username (e.g., "@jack"). In both cases, the user being referred to, whether through a retweet or mention, receives notifications from the service. The number of retweets typically reflects the content's value in a tweet, while the number of mentions is associated with the user's name recognition or fame. These interactions play a role in assessing the significance of content and user influence on Twitter.

2.9 Performance Evaluation

Various metrics are available for assessing the performance of Data Mining or Machine Learning classifiers. These metrics rely on a "Confusion Matrix," [54] which includes the following components:

- True Positives (TP): These are instances correctly predicted as positive.
- True Negatives (TN): These are instances correctly identified as negative.
- False Positives (FP): These are instances incorrectly predicted as positive, typically occurring when a comment is inaccurately labeled as cyberbullying behavior.
- False Negatives (FN): These are instances incorrectly labeled as negative when they should have been labelled as positive.

The performance metrics utilized in this study are described as follows:

2.9.1 Over all Accuracy

Accuracy: This widely used metric is defined as the ratio of correctly predicted values to the total values and is calculated using the formula:

$$\text{Accuracy}=\frac{TP+TN}{TP+FP+TN+F} \tag{2.13}$$

2.9.2 Precision

Precision: Precision measures the proportion of relevant observations that are correctly predicted as positive values out of all the predicted positive values and is calculated as:

$$\text{Precision}=\frac{TP}{TP+FP} \tag{2.14}$$

2.9.3 Recall

Recall: Recall represents the ratio of correctly returned relevant values to the total values in the entire class and is determined by the formula:

$$\text{Recall}=\frac{TP}{TP+FN} \tag{2.15}$$

2.9.4 F1-measure

F1-measure: The F1-measure is a particular case of the F-measure and serves as the weighted mean of Precision and Recall. It addresses the issue of the negative correlation between Precision and Recall. The formula for the F1-measure is given by:

$$\text{F1-measure}=\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Rec}} \qquad (2.16)$$

Additionally, the F1-measure introduces a parameter β to control the balance between Recall and Precision, where $0 \leq \beta \leq \infty$.

CHAPTER THREE

PROPOSED SYSTEM

3.1 Overview

This chapter provides the main content of the proposed system. First, discuss dataset statistics and the pre-processing step. Secondly as shown in figure 3.1 proposed model stages.

3.2 Proposed System

The proposed system is Twitter CBD.SNG is divided into two models each model consists of steps to achieve thesis objectives. The first model takes a conventional approach, primarily focusing on text analysis. This phase involves a sequence of five steps: preprocessing, feature extraction, feature selection, partitioning of data for training within the classification models and, evaluation. As shown in figure 3.1. Detailed explanations for each of these steps will be provided later.

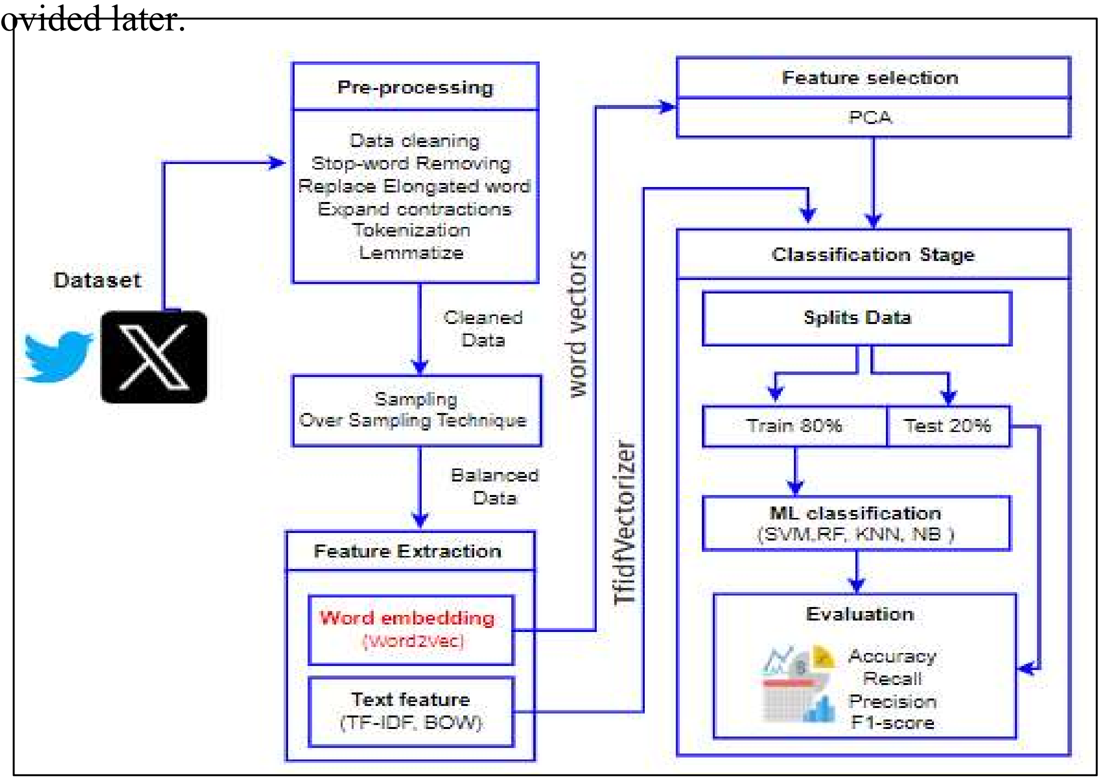


Figure 3.1: Model 1: Cyberbullying Classification

In the second model, Social Network Analysis (SNA) utilized to identify individuals who participate in bullying behavior (influential nodes) in new ways. By building co-occurrence patterns amongst users, proposed system aim to improve the classification of bullies within the graph. This is achieved through analyzing tweets to discover common elements, such as mentions and hashtags commonly used in these tweets.

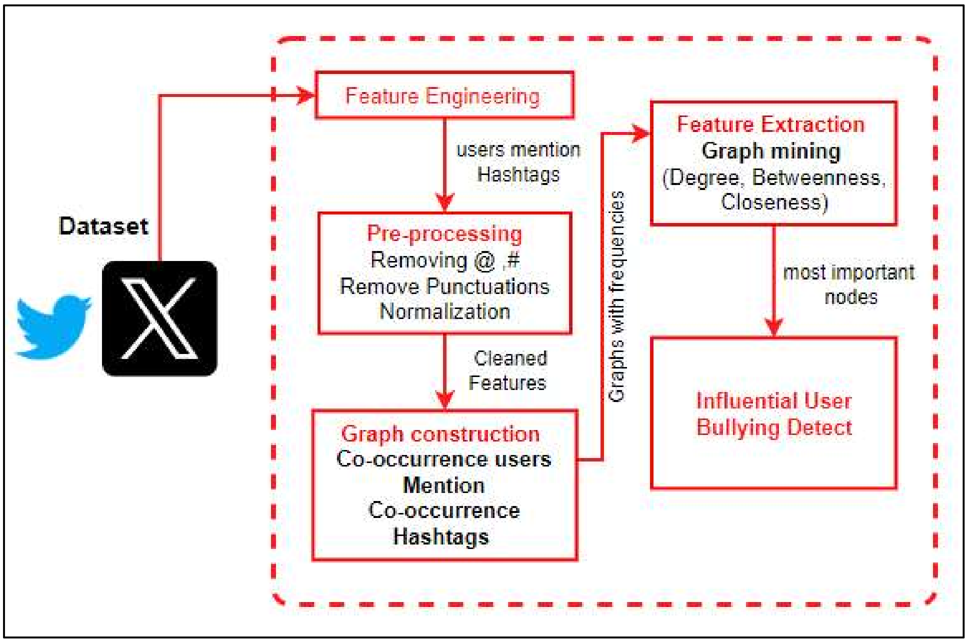


Figure 3.3 : Model 2: Cyberbullying Detection

3.3 Data Pre-processing

The data must be processed to ensure that it has been cleaned and is prepared for analysis. The raw tweets' content contains URLs that start with http or https. They also contain emojis such as smile face and grinning face. Also, there are many punctuation and spaces between words in order to make it clean, the following steps were applied. After completing all steps mentioned in section

2.3, the results are cleaned, normalized, and tokenized version of the text ready for analysis and modeling.

3.3.1 Resampling step

Both datasets started in a balanced state, but during the pre-processing steps, including removing duplicated tweets and those with only two words, the datasets became imbalanced; this is a common issue in classification tasks. To address this challenge, widely accepted resampling methods are employed, which involve augmenting the records in the minority class. As discussed in Chapter Two, Section 2.3.5, two primary techniques for addressing class imbalance are undersampling and oversampling. In this thesis, oversampling used, which to increases the representation of the minority class by replicating data points. Ensuring dataset balance is crucial as it guarantees that machine learning models treat both classes fairly, ultimately improving model performance.

3.4 Feature Extraction

Feature extraction refers to extracting and presenting feature representations that are convenient for the task needed to be accomplished and the type of model planned to be built. It is performed to extract meaningful features or attributes from textual tweets to be ready for the classification method. To classify text tweets using machine learning algorithms, they need to work on numerical vectors only as they are unable to use raw text data that has formats that obstruct the work of these algorithms. Cyberbullying text categorization requires feature extraction. Our model extracts features using TF-IDF, Word2Vec, and graph mining features.

3.4.1 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a score focused on each relevant word in the tweet. The term frequency TF is the number of repetitions of a word in a tweet divided by the total number of words in the same tweet, while the inverse document frequency is the logarithm of the number of tweets divided by the number of tweets containing the word. The steps of finding the importance of a word and assigning a weight to it are as follows:

1. Use equation (2.1) to find the TF for each word.
2. Use equation (2.2) to find the IDF for the same words.
3. Vectorization of the vocabulary.

3.4.2 Pre-Trained Word Embeddings

Pre-trained word embeddings are embeddings that have been learned in one task and may be utilized to solve another comparable task. As they are taught on huge datasets, pre-trained word embeddings capture a word's semantic and syntactic meaning. They have the ability to improve an NLP model's performance.

3.5 Classification Step

Four distinct machine learning models were applied in supervised multi-classification to categorize 49,656 tweets across six classes using the

Cyberbullying Classification Dataset. Additionally, 2,108 tweets spanning five classes were classified using the Cyber Bullying Types dataset. In both datasets, the data was partitioned into separate subsets for training, validation, and testing. Specifically, 80% of the data was allocated for training purposes, within which a further 60% was dedicated to the primary training set, and the remaining 20% served as the validation set. The remaining 20% of the data was preserved for testing the models' performance. This approach allowed for rigorous evaluation and validation of the machine learning models' effectiveness in cyberbullying detection and classification tasks. Table 3.1 provides a selection of example tweets and their corresponding cyberbullying types, sourced from the "Cyberbullying Classification Dataset (CBCD)." In contrast, Table 3.2 presents sample tweets and their respective classes derived from the "Cyber Bullying Types Dataset (CBTD)."

Table 3.1 Sample Cyberbullying Classification Dataset

Tweet text	**Cyberbullying type**
"I hear that it's snowing up north. Glad I made it through that before the snow started."	not_cyberbullying
"@jamuraa yupppp. jason doesn't have a lot of experience on twitter, it seems. or a very healthy world view."	other_cyberbullying
"@DeeSaysTheTruth F*ck You Dumb Nigger"	ethnicity
"@DavidHarvilicz @AFP More left wing scum caving in to the violence of Islam."	religion
"Because gay jokes and rape jokes are really not at all funny."	gender
"If you want to be a school board lawyer, it helps to be a bully"	age

Table 3.2: Sample of Cyber Bullying Types Dataset

Tweet	Class
“Post-Cuomo push to toughen NY sexual harassmentÂ laws https://t.co/BgiuSqIbAe	Sexual Harassment
Woman claims internet gave her PTSD and it's as serious as war veterans https://t.co/Y9WsbyaLRn via @MailOnline ðŸ¤¦ðŸ »â€ â™‚ï¸	Cyberstalking
“I don't wanna be a Republican and I don't wanna be a Democrat. I wanna be a goddamn American and I wanna listen toâ€¦ https://t.co/uEXGkE5hV9”	Doxing
“Revenge porn will land your butt in prison.”	Revenge Porn
“Slut shaming friends... who do exactly what they be judging other women for ! https://t.co/P6FeHTmWnf”	Slut Shaming

These tweets were used to construct thesis first model. Four models were applied to find the best one, these are:

- Support Vector Machine (SVM) model
- Random Forest (RF) model
- K-Nearest Neighbours (KNN) model
- Multinomial Naïve Bayes (Multinomial NB) model

After applying the four models, it was found that Random Forest (RF) outperformed others in both datasets. But the output of CBCD is better, therefore, we based on it to make social network analysis (SNA).

3.6 Social Network Analysis Step

Social network analysis techniques encompass two main elements: data mining and social network analysis. The selection and extraction of data features constitute a pivotal initial step in cyberbullying detection. Social network attributes, such as the proximity between users and the overall aggressiveness of the social network, can also influence the results of cyberbullying classification. Thesis experiments introduces new features from the CBCD dataset, namely user mentions and hashtags. In the following figure 3.4 illustrates process users mention network analysis and figure 3.5 illustrate Hashtags Network Analysis Process

As outlined in the below algorithm 3.1, this step employ a distinct new features extraction (user mentions and hashtag), then making pre-processing for the list of hashtags "#" and removing punctuation marks and symbols. Additionally, the hashtag symbol itself is eliminated from the resulting list.

Following removing the "@" symbol and cleaning the list of mentioned users by removing symbols and linguistic punctuation, the refined list is obtained.

Algorithm 3.1: Features Extraction

```
Input: Tweet_text(t)
Output: Clean Hashtags feature (a) , Clean Mentions feature (b)
Begin
        Step 1: h_list = re.findall(r'#\w+', str(t))
          Step 2: for h in h_list
             Step 2.1: a =  re.sub(r'^#', '', h.lower())
                   end for
          Step 3: m_list = re.findall(r'@(\w+)', str(t))
          Step 4: for m in m_list
             Step 4.1: b =  re.sub(r'@(\w+)', m.lower())
                   end for
```

End

The next step involved the implementation of two distinct procedures, constituting an important component within the framework of social network analysis. It is the graph construction.

Through algorithm 3.2, co-occurrence mentions. The concept of Clean Mentions was addressed, and a compilation of non-empty Clean Mentions will be generated. Specifically, the objective is to establish a list denoted as non-empty Clean Mentions, wherein mention1 is included as a constituent. When iterating through the non-empty Clean Mentions, it is necessary to verify that mention1 is not equivalent to mention2. Construct a graph representing the co-occurrence mentions the cosine similarity method apply as edge weights between each pair of mentions. Calculate the frequency of each mention.

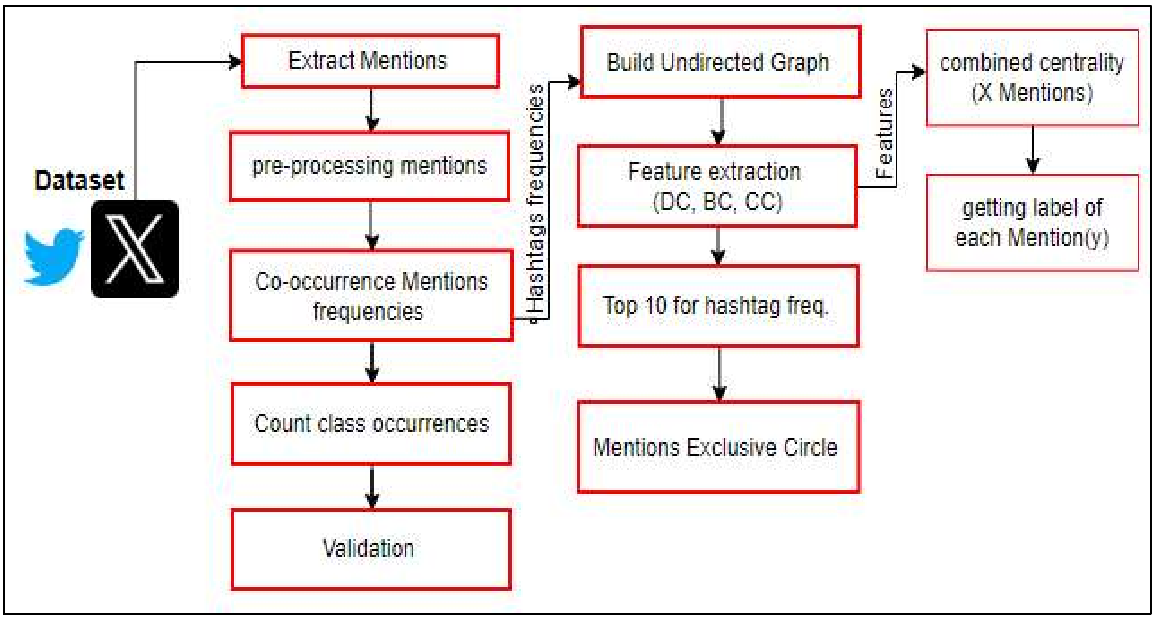

Dataset
Extract Mentions
pre-processing mentions
Co-occurrence Mentions frequencies
Count class occurrences
Validation
Hashtags frequencies
Build Undirected Graph
Feature extraction (DC, BC, CC)
Top 10 for hashtag freq.
Mentions Exclusive Circle
Features
combined centrality (X Mentions)
getting label of each Mention(y)

Figure 3.4: Users Mention Network Analysis Process

Algorithm 3.2 Mention Graph Construction

```
Input: Clean Mentions(cm), mention(m)
Output: Co-occurrence Mentions Graph (GM), mention frequency (m_freq), edge
weights (w)
Begin
    Step 1: for m in cm
                x = [ m for m in m_list if m.strip() != ‘’]
                  for m1in x
                    for m2 in x
                      if m1 != m2
                        G.add_edge[(m1, m2)]
                        m_freq[m1] += 1
                      end if
    Step 2: Return GM, m_freq
    Step 3: for i in range(len(m)):
                for j in range(i + 1, len(m)):
                  m1, m2 = m[i], m[j]
                  sim =CosSimilarity(m1, m2)
                  w[(m1, m2)] = sim
    Step 4: Return w
End
```

In algorithm 3.3, after constructing the graph representing user mentions and calculating their frequency counts, the subsequent step involves sorting these counts to identify the top user mentions. These users are considered prominent figures in the context of cyberbullying. The proposed model analyze to ascertain whether a user qualifies as a bully or belongs to other categories of cyberbullying prevalent in our dataset. Specifically, in this thesis calculated the occurrence count of each cyberbullying type associated with these top user mentions. This comprehensive approach allows us to validate the categorize users concerning their engagement in cyberbullying activities.

Algorithm 3.3: Analysis Mention label occurrence

```
Input: Mention frequency(m_freq) , mention(m), mention label(y)
Output: Mention label counts(cs), Mention label length(m_len)
Begin
            Step 1: m_sorted = sorted(m_freq())
            Step 2: m_top=list(m_sorted)[0:10])
            Step 3: for m in m_top
                        c = Counter(y)
                        cs[m] = c
                        m_len = len(y)
            Step 4: for m, c_and_len in cs.item():
                        c= c_and_len[c]
                        m_len= c_and_len[len]
            Step 5: return cs ,m_len
 End
```

Algorithm 3.4 takes the cleaned user mentions obtained from algorithm 3.1 and constructs a user mentions graph to evaluate the significance of each user within the network. This assessment is based on three fundamental centrality measures: degree centrality, betweenness centrality, and closeness centrality, each offering unique insights into the network's dynamics.

Degree centrality provides a gauge of a user's popularity or prominence within the network, quantifying their engagement level and interactions with others. Meanwhile, betweenness centrality identifies users who serve as pivotal connectors or intermediaries in the network, exerting substantial influence over the flow of information. Lastly, closeness centrality measures how efficiently users can reach others within the network.

These centrality metrics collectively enhance our understanding of the network's underlying structure and dynamics. Empower us to pinpoint critical influencers, recognize network substructures, and make well-informed decisions.

Algorithm 3.4: Mention Graph Construction

```
Input: Mention Graph (GM)
Output: Mention exclusive circle(y)
Begin
        Step 1: DC = degree_centrality(GM)
                DC_sorted = sorted(DC)
                DC_top=list(DC_sorted)[0:10])

        Step 2 BC = betweenness_centrality(GM)
                BC_sorted = sorted(BC)
                BC_top=list(BC_sorted)[0:10])

        Step 3: CC = closeness_centrality(GM)
                CC_sorted = sorted(CC)
                CC_top=list(CC_sorted)[0:10])

        Step 4: y = set(DC_top) | set(BC_top) | set(CC_top)
End
```

Through algorithm 3.5, co-occurrence hashtags, a list of non-empty Clean Hashtags was generated and iterated through each hashtag in the collection of

Algorithm 3.5: Hashtag Graph Construction

```
Input: Clean Hashtags (a), hashtag (h).
Output: Co-occurrence Hashtag Graph(G_hash), hashtag frequency(h_freq)
Begin
        Step 1: for h in a
          Step 1.1: x = [ h for h in h_list  if h.strip() != '']
          Step 1.2: for h1 in x
             Step 1.2.1: for h2 in x
                         if h1 != h2
                              G.add_edge(h1, h2)
                           h_freq[h1] += 1
                         end if
        Step 2: Return G_hash, hash_freq
End
```

Clean Hashtags. When iterating through the non-empty Clean Hashtags, denoted as hashtag1, and the non-empty Clean Hashtags, denoted as hashtag2, it is necessary to verify that hashtag1 is not identical to hashtag2 to construct a graph.

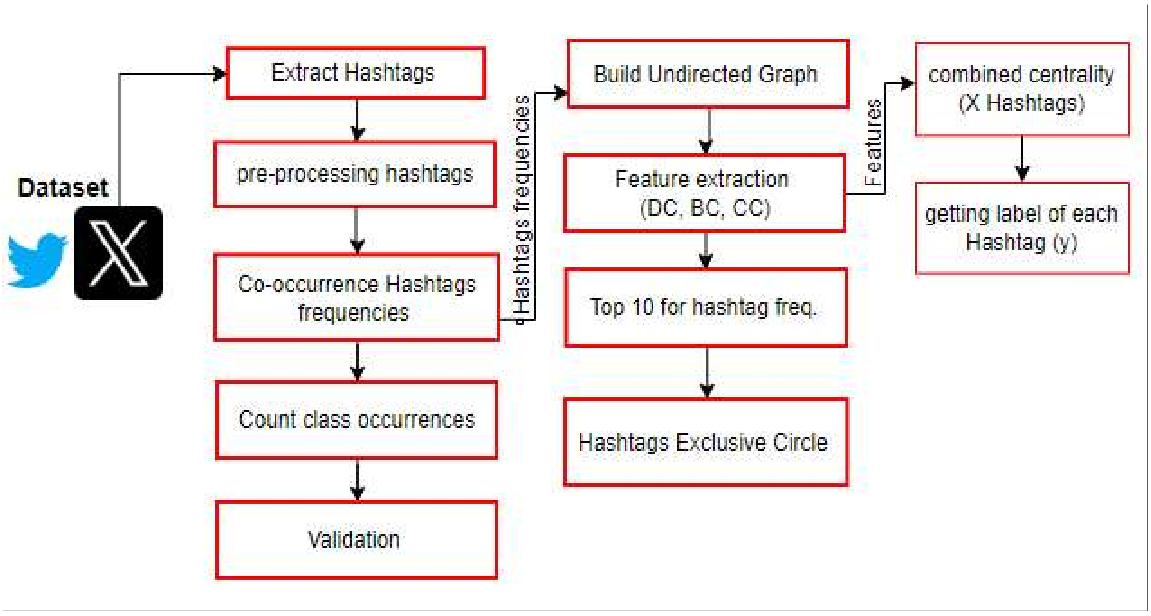

Dataset
Extract Hashtags
pre-processing hashtags
Co-occurrence Hashtags frequencies
Count class occurrences
Validation
Hashtags frequencies
Build Undirected Graph
Feature extraction (DC, BC, CC)
Top 10 for hashtag freq.
Hashtags Exclusive Circle
Features
combined centrality (X Hashtags)
getting label of each Hashtag (y)

Figure 3.5: Hashtag Network Analysis Process

Algorithm 3.6, which follows the creation of the hashtags graph, is dedicated to arranging the usage of hashtags into a frequency count analysis. The primary objective is to determine the frequency of each of the top 10 hashtags, often called "Top-Hashtags." For each Top-Hashtag, the algorithm systematically examines its usage, recording associated label occurrences. It then calculates the aggregate count of all labels associated with the current hashtag. This collective information assists in identifying the type of topic corresponding to each hashtag. The results are meticulously stored in a dictionary called "Hashtag Label counts." This step represents a crucial part of the system's allows us to validate the categorize hashtag concerning it's engagement in cyberbullying topics.

Algorithm 3.6: Analysis Hashtag label occurrence

```
Input: Hashtag frequency(h_freq) , hashtag(h), hashtag label(y)
Output: Hashtag label counts(cs), Hashtag label length(h_len)
Begin
        Step 1: h_sorted = sorted(h_freq())
        Step 2: h_top=list(h_sorted)[0:10])
        Step 3: for h in h_top
          Step 3.1:   c = Counter(y)
          Step 3.2:   cs[h] = c
          Step 3.3:    h_len = len(y)
        Step 4: for h, c_and_len in cs.item():
          Step 4.1:    c= c_and_len[c]
          Step 4.2:    h_len= c_and_len[len]
        Step 5: return cs ,h_len
End
```

Algorithm 3.7 utilizes cleaned hashtags from Algorithm 3.1 to construct a co-occurrence-based hashtags graph, enabling the evaluation of hashtag significance within the network. Using DC, BC, and CC provides valuable insights into hashtag analysis by understanding users' influence based on hashtag interactions and identifying critical players within the network. Influence Analysis assesses users' impact on hashtag adoption.

Then, get the top ten of each DC, BC, and CC, create an exclusive Circle of hashtags and get the class of each hashtag.

Algorithm 3.7: Hashtag Graph Construction

```
Input: Hashtag Graph (G_hash)
Output: Hashtag exclusive circle(y)
Begin
        Step 1: DC = degree_centrality(G_hash)
                DC_sorted = sorted(DC)
                DC_top=list(DC_sorted)[0:10])

        Step 2 BC = betweenness_centrality(G_hash)
                BC_sorted = sorted(BC)
                BC_top=list(BC_sorted)[0:10])

        Step 3: CC = closeness_centrality(G_hash)
                CC_sorted = sorted(CC)
                CC_top=list(CC_sorted)[0:10])

        Step 4: y = set(DC_top) | set(BC_top) | set(CC_top)
End
```

3.7 Performance Evaluation

Since cyberbullying detection is considered as a classification problem, so can apply different classification evaluation methods i.e. Precision, Recall, F1-score and Accuracy (explained in section 2.9).

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Overview

This chapter discusses the experimental and achieved results from each phase of the proposed system, which utilizes two primary datasets: the Cyberbullying Classification Dataset (CBCD) and the Cyber Bullying Types Dataset (CBTD), as previously introduced in Chapter Three. Additionally, it presents the newly extracted features from the CBCD. The chapter details the hardware and software necessary to implement the proposed system successfully.

4.2 Software and Hardware

The proposed system was implemented using the following hardware and software requirements.

Hardware: Processor Intel i7, RAM 16GB, Storage 512 GB, Freq. 2.60GHz.

Software: Operating System: Windows 10 pro-64-bit.

Programming language: Python

4.3 Datasets

The Twitter platform has provided helpful information for text analysis, such as tweet contents and user information. Also, it is one of the most popular resources used in the cyberbullying analysis. There are two datasets used in this Thesis:

4.3.1 Cyberbullying Classification Dataset

The Cyberbullying Classification was made by J. Wang, K. Fu, and C.T. Lu [1] . The file was downloaded from Kaggle, and it is a balanced dataset. It has 47692 entries, as shown in Table 1. and two columns: "tweet_text" and "cyberbullying_type." It also had six categories: "not_cyberbullying," "other_cyberbullying," "ethnicity," "religion," "gender," and "age". Figure 4.1 illustrate count of each class in **CBCD.**

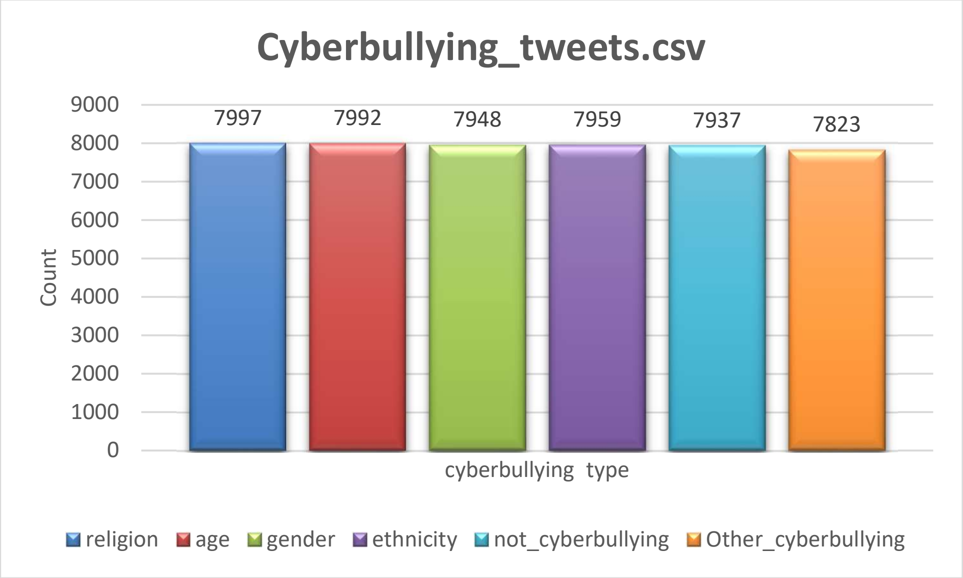


Figure 4.1: Cyberbullying Classification Dataset

4.3.2 Cyber Bullying Types Dataset (CBTD)

Cyber Bullying Types Datasets was created by Dr. N. Anathi[2]contains 2140 entries. It has two columns and five categories for cyberbullying, including

[1] https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification

[2] https://ieee-dataport.org/documents/cyber-bullying-types-datasets#

sexual harassment, doxing, cyberstalking, revenge porn, and slut shaming. The first column represents textual content, while the second column indicates the category of cyberbullying. The dataset was downloaded from IEEE Dataport and its balanced dataset. Figure 4.2 illustrate count of each class in **CBTD.**

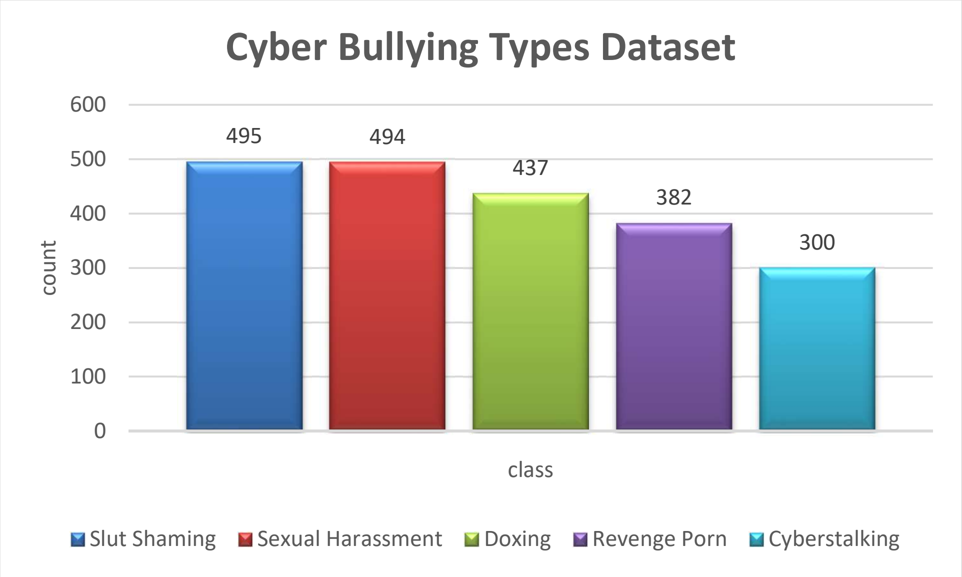


Figure 4.2: Cyber Bullying Types Dataset

4.4 Result of Data Pre-Processing

Results shown in the table 4.1 below after making pre-processing steps explained in chapter two in section 2.3 (Remove duplicates, Remove Numbers, Remove Punctuation, Remove Whitespaces, Eliminate characters, Correct Contractions, Lowercasing, Replace Elongated Words, Text Tokenization, Text Normalization, Stop-Word Removal and Eliminate Specific Entities).

Table 4.1: Example of Datasets before and after pre-processing

Dataset name	Before pre-processing	After pre-processing
Cyberbullying Classification Dataset	“@XochitlSuckkks a classy whore? Or more red velvet cupcakes?”	classy whore red velvet cupcake
Cyber Bullying Types Dataset	“@Fr0gK1ng If you ever post this again I am doxing you”	ever post doxing

Although the cyberbullying classification dataset, as mentioned above, is balanc ed, there are around 4141 duplicate tweets, which will re-moved. The following Figure 4.3 illustrate counts of each class after removed duplicate tweets:

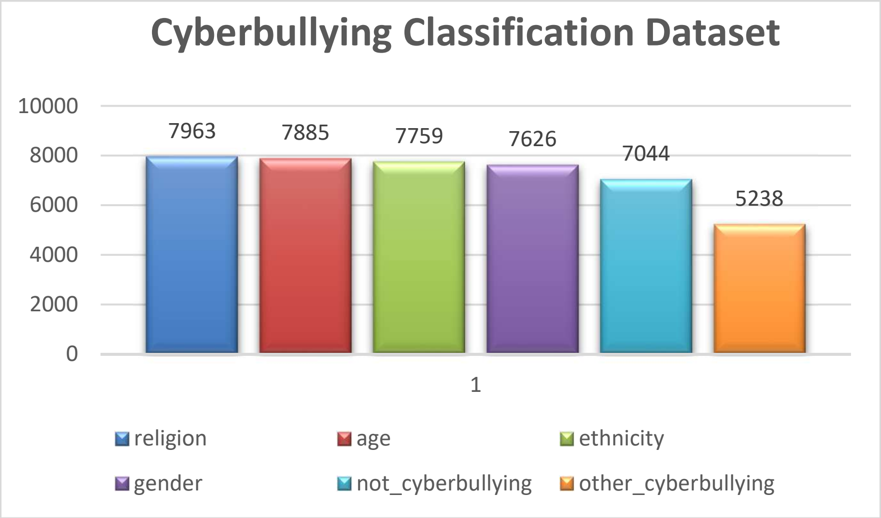


Figure 4.3: CBCD after removing duplicate tweets

So there is a problem appear the classes be unequal distribution imbalanced, so to solve this problem using Resampling oversample the training set so that each class has the same number of members as the class with the highest population.

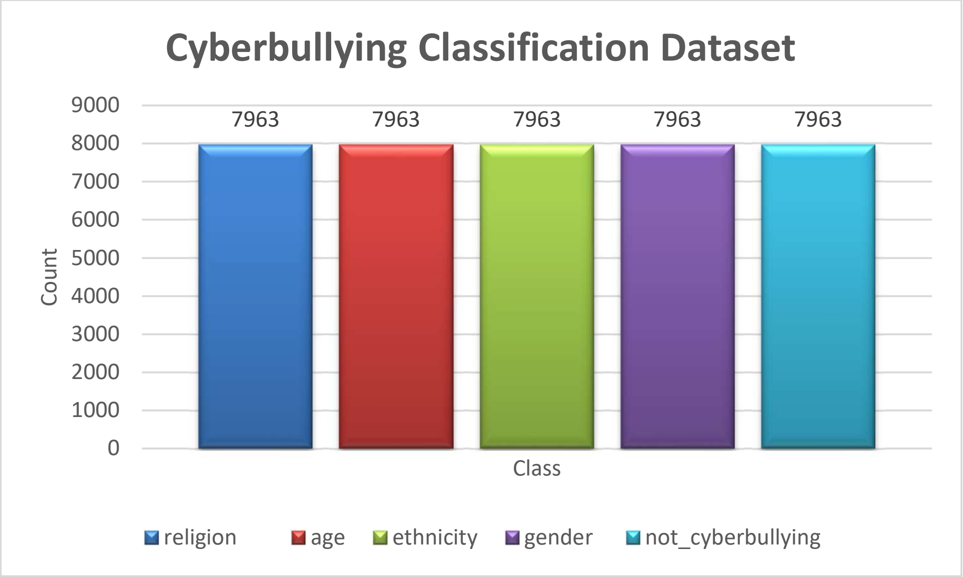


Figure 4.4: CBCD after Oversampling

The CBTD had around 86 duplicate tweets; thus, the following classes count after removing duplicate tweets:

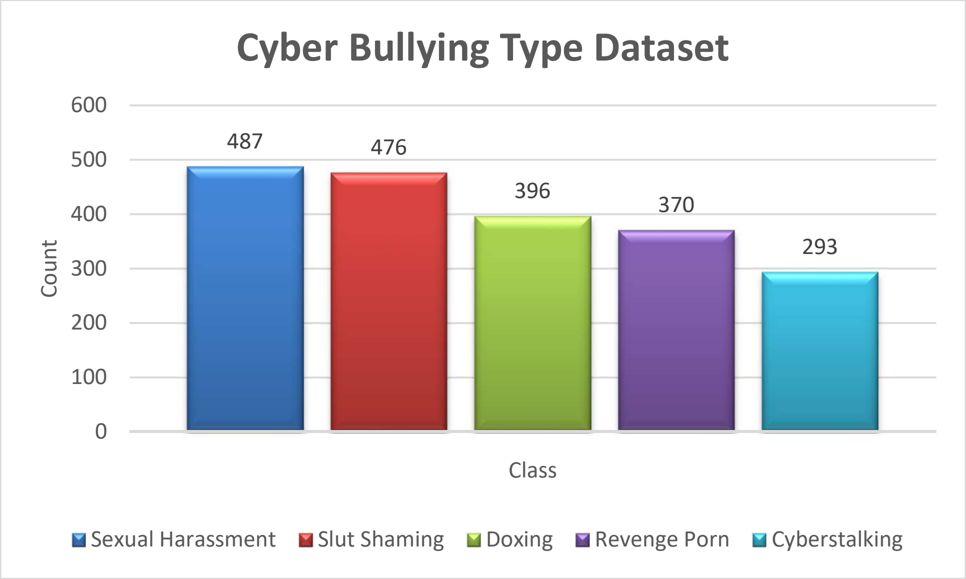


Figure 4.5: CBTD after removing duplicate tweets

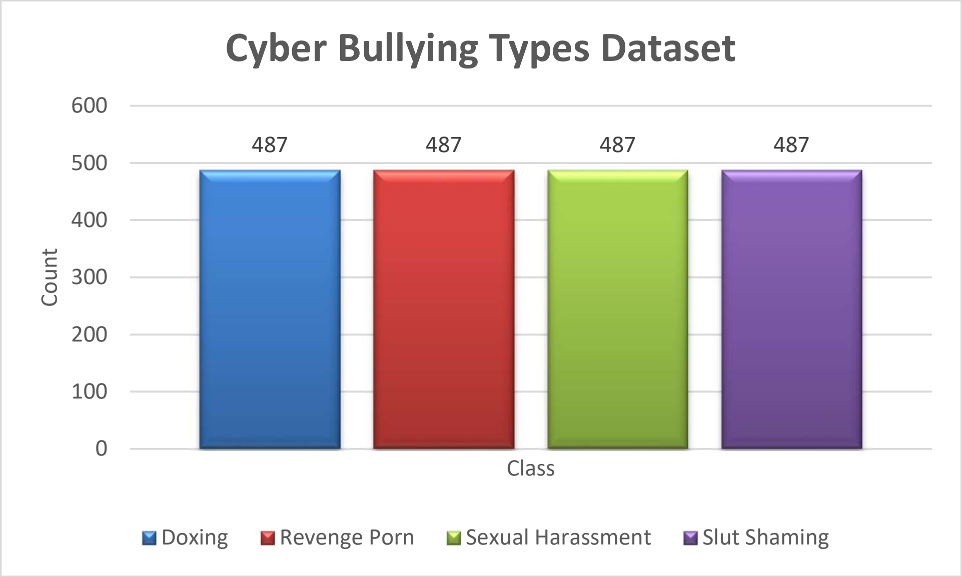


Figure 4.6: CBTD after Oversampling

4.5 Feature Extraction Results

After processing the data, it is crucial to extract features from the tweet content and use these features to analyse whether a tweet's content qualifies as cyberbullying or not.

4.5.1 TF-IDF and BOW Results

The table below illustrates the performance metrics of various classifiers with the combination of TF-IDF and CountVectorizer feature extraction methods. Notably, the SVM and RF classifier's achieved an impressive accuracy of 93%, along with high precision, recall, and F1-score values, all at 0.93. These results indicate the robustness of the combination in accurately classifying data points. The NB and KNN classifiers perform slightly lower accuracy at 84% and 86%, respectively, but maintain a good balance of precision, recall, and F1-score. Overall, this combination demonstrates its competence in text classification tasks, with SVM and RF standing out as top-performing options, especially when high accuracy and precision are required.

Table 4.2: Performance Metrics Using TF-IDF/CountVctorizer in CBCD

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-Score**
SVM	93%	0.93	0.93	0.93
NB	84%	0.85	0.84	0.83
RF	**93%**	0.93	**0.94**	0.93
KNN	86%	0.86	0.86	0.85

Table 4.3: Performance Metrics Using TF-IDF/CountVctorizer in CBTD

Classifier	**Accuracy**	**Precision**	**Recall**	**F1- Score**
SVM (RBF)	87%	0.88	0.87	0.87
NB	84%	0.84	0.84	0.84
RF	**89%**	**0.90**	0.89	0.89
KNN	85%	0.86	0.85	0.85

4.5.2 Word2Vec Results

In this thesis, to convert tweet content into a feature vector, we compute the average feature vector for words present in the tweet and match them with a vocabulary list generated by a trained Word2Vec model using the entire tweet dataset. It's important to consider that the model's performance can be impacted by different parameter values.

Table 4.4: *Performance Metric for the Word2Vec in CBCD*

Classifier	Accuracy	Precision	Recall	F1-Score	Time (seconds)
SVM (poly)	86%	0.87	0.86	0.86	11.89
KNN	82%	0.86	0.82	0.82	0.01
RF	86%	0.86	0.86	0.86	21.74
NB	31%	0.34	0.31	0.34	0.28

For experiment, each classifier's performance on the cyberbullying classification dataset using Word2Vec embeddings:

SVM (Poly): The Support Vector Machine with a polynomial kernel demonstrates robust performance with high accuracy, precision, recall, and F1-Score, making it a strong candidate for cyberbullying classification. However, it requires a relatively longer training time, which could be a trade-off for accuracy and effectiveness.

K-Nearest Neighbors (KNN): KNN provides reasonable performance metrics with good precision, though its accuracy and recall are slightly lower than SVM and LR. The critical advantage of KNN is its rapid training time, making it suitable for real-time or resource-constrained applications.

Random Forest (RF): RF yields high accuracy and F1-Score at the cost of longer training times. This classifier excels when computational resources are abundant, delivering robust performance in exchange for the additional time required.

Naive Bayes (NB): demonstrates the poorest performance among the classifiers, with low accuracy, precision, recall, and F1-Score. It is not recommended for this dataset due to its inability to handle the intricacies of cyberbullying classification.

Table 4.5: *CBTD the Performance Metric for the Word2Vec*

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Time (seconds)**
SVM (poly)	%86	0.87	0.86	0.86	14.66
KNN	%83	0.87	0.83	0.83	0
RF	%85	0.86	0.85	0.85	22.55

The performance of each classifier on the CBTD using Word2Vec embeddings for multiclassification discussion is as below:

SVM (Poly): SVM with a polynomial kernel in multiclassification for cyberbullying detection, achieving a good accuracy of 86% and balanced precision, recall, and F1-Score, each at 0.86. Its robust performance comes at the cost of a longer training time (14.66 seconds). This classifier is well-suited for handling complex multiclass scenarios, offering reliable results across different classes.

K-Nearest Neighbors (KNN): performs well in multiclassification with an accuracy of 83% and balanced precision, recall, and F1-Score at 0.83. What sets KNN apart is its rapid training time (0 seconds). This classifier is an excellent choice for quick and reliable multiclass classification results.

Random Forest (RF): provides multi-classification solid performance with an accuracy of 85% and balanced precision, recall, and F1-Score at 0.85. However, it comes with a longer training time (22.55 seconds). This classifier is an excellent choice when accuracy is a priority and computational resources are not constrained.

In conclusion, the choice of the best classifier for multiclass in the CBTD depends on a trade-off between performance, training time, and resource availability. SVM (Poly) and RF offer high accuracy, while KNN balance performance and efficiency.

4.5.3 Graph mining

Leveraging the advantages of Word2Vec embeddings, cosine similarity employed as the metric to construct a weighted graph, depicted in the figure 4.7. This graph is built by measuring the cosine similarity between word vectors, allowing us to establish the strength of relationships between words based on their semantic similarities. The resulting graph is a valuable tool for exploring semantic relationships and insights within the dataset, enhancing our ability to uncover meaningful connections and patterns.

Cause of big data to visualize the nodes choosed just 50 node with similarity greater than 0.971. in the table *4.6* illustrate the highest cosine-similarity.

Table 4.6: *Top 10 Nodes Cosine Similarity*

node	**CosSim**
lgbt	0.98
holocaust	0.98
equates	0.98
mocking	0.98
gang	0.98

marriage	0.97
harm	0.97
bashing	0.97
abuse	0.97
pedophile	0.97

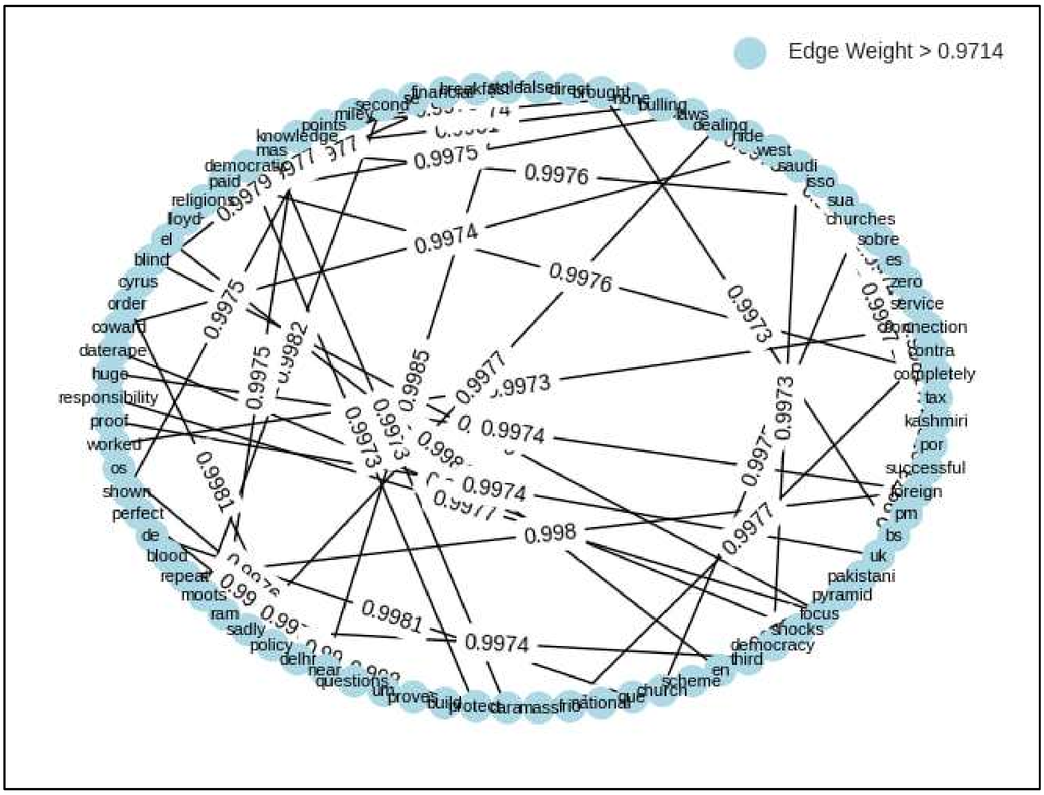


Figure 4.7: Visualization Network Graph of CBCD

Centrality refers to a set of measures that assess the significance of a node within a network. Various methods employed to calculate centrality, but our emphasis will be on three critical approaches: degree centrality, closeness centrality, and betweenness centrality.

Table 4.7: *Top 10 Nodes DC, BC and CC in CBCD*

seq.	node	DC	node	BC	node	CC
1	contra	0.039	contra	0.007	contra	0.059
2	bulling	0.026	bulling	0.003	bulling	0.038
3	count	0.026	count	0.001	listening	0.03
4	listen	0.026	listen	0.001	rock	0.02
5	truth	0.013	truth	0.001	truth	0.01
6	agenda	0.013	agenda	0.001	major	0.01
7	spread	0.013	spread	0.001	chinese	0.01
8	greek	0.013	greek	0.001	information	0.01
9	random	0.013	random	0.001	favorite	0.01
10	avoid	0.013	avoid	0.001	arabia	0.01

In the table 4.10 used top 10 when visualizing the CBCD, 21 observed nodes with a cosine similarity score exceeding 0.971. These nodes are depicted in Figure 4.8 shows the cosine similarity values for each item listed alongside it.

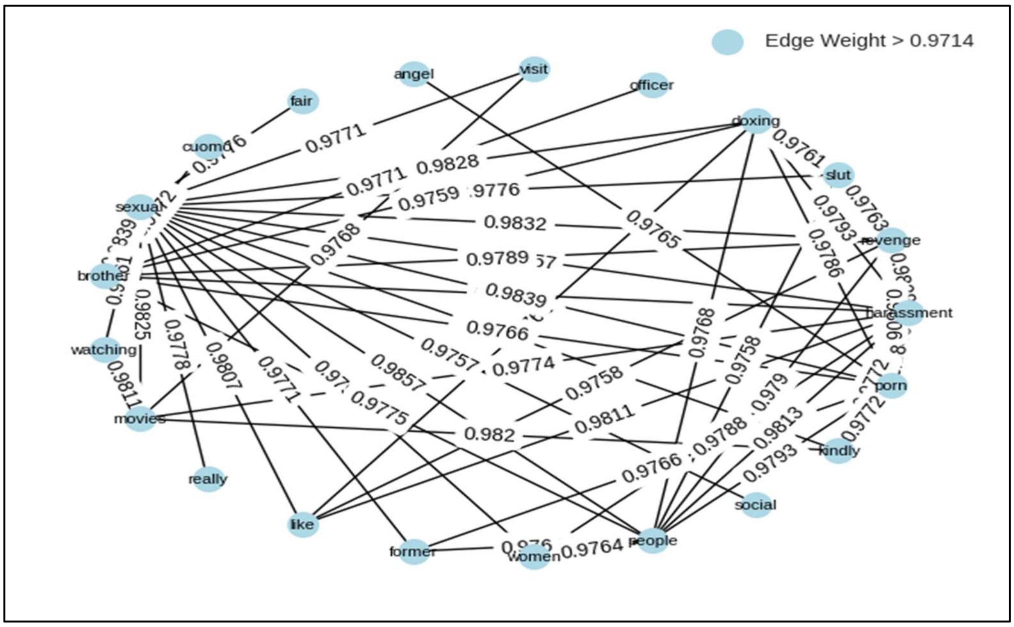


Figure 4.8: Visualization Network Graph of CBTD

On the other hand, when visualizing the CBTD, 21 observed nodes with a cosine similarity score exceeding 0.971. These nodes are depicted in Figure 4.8 shows the cosine similarity values for each item listed alongside it.

As shown in table 4.8, node 'Sexual' emerges as the most central node across all three measures, signifying its pivotal role in the network. 'Harassment' and 'people' also maintain significant centrality in each category. These results highlight the importance of these terms within the context of cyberbullying in the dataset, suggesting their prevalence, influence, and potential significance in understanding cyberbullying dynamics. Closeness centrality, in particular, underscores the efficiency of these nodes in connecting with others, further emphasizing their central positions in the network.

Table 4.8 *Top 10 Nodes DC, BC and CC in CBTD*

Seq.	Node	DC	Node	BC	Node	CC
1	Sexual	0.85	sexual	0.58	sexual	0.86
2	harassment	0.5	brother	0.27	harassment	0.66
3	People	0.45	porn	0.2	people	0.64
4	Porn	0.45	harassment	0.15	porn	0.64
5	brother	0.4	people	0.13	brother	0.62
6	revenge	0.35	revenge	0.1	revenge	0.6
7	doxing	0.35	doxing	0.1	doxing	0.6
8	movies	0.25	movies	0.1	movies	0.1
9	slut	0.2	kindly	0.09	kindly	0.52
10	like	0.2	slut	0.095	slut	0.51

4.6 Social Network Analysis Results

Our experiment requires further details about the social network and communication dynamics. To gain a more comprehensive view of user relationships, the original dataset have expanded by including supplementary features, as depicted in the table below.

Table 4.9: *CBCD after Adding New Features*

Text	Class	Hashtags	Mentions
"RT @peteevansnot: Seriously why wouldn't you feed #paleo formula to your newborns with TWENTY TIMES the vitamin A of breastmilk? #mkr http:â€¦"	not_cyberbullying	['#paleo', '#mkr']	['@peteevansnot']
"REALLY wish it were b*tch face and her husband in sudden death ðŸ˜¡ #mkr @mykitchenrules"	gender	['#mkr']	['@mykitchenrules']
"#stopwadhwa2015 @theonion wrote an article about @wadhwa. http://t.co/DlN25sa74H (via @bartitos)"	other_cyberbullying	['#stopwadhwa2015']	['@theonion', '@wadhwa', '@bartitos']

4.6.1 Users Mention

After extracting new feature user mentions (UM) from the dataset and performing necessary preprocessing, the frequency of each UM was computed to identify the most frequently mentioned. As shown in table 4.10.

Table 4.10: *Top 10 Mention Frequencies*

seq.	Mention	Frequency
1	'freebsdgirl'	481
2	'tayyoung_'	404
3	'BilalIGhumman'	265
4	'MT8_9'	261
5	'IsraeliRegime'	242
6	'MaxBlumenthal'	241
7	'greenlinerzjm'	219
8	'98Halima'	195
9	'johnnygjokaj'	195
10	'srhbutts'	192

This analysis allowed us to classify the user with the highest mention frequency as the influential user.

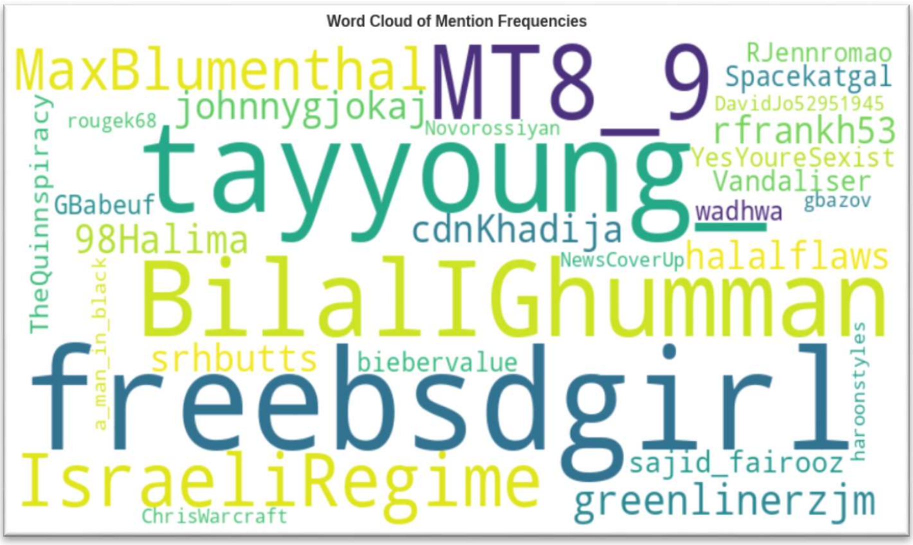


Figure 4.9: Word Cloud of Users Mentions Frequency

The decision to designate a specific user as the 'Bully' was grounded in a thorough analysis of their UM, which exhibited the highest mention frequency in the dataset, signifying their influential presence. This labelling aimed to accentuate their central role within the dataset and their potential impact on interactions. To Validate this classification, an in-depth investigation was conducted, quantifying the UM's frequency across different classes, as detailed in the table 4.11. This approach reinforced the ‘Bully’ classification and emphasized their significance in shaping the dataset’s discussions and interactions.

Table 4.11: *Top 10 Mentions with Its Class Counts*

		Classes				
N	**Mention**	**other_cyberbullying**	**gender**	**ethnicity**	**religion**	**not_cyberbullying**
1	‘freebsdgirl’	149	7	0	0	75
2	‘tayyoung_’	0	0	957	0	0
3	‘BilalIGhumman’	0	0	0	46	27
4	‘MT8_9’	5	89	0	0	7
5	‘IsraeliRegime’	0	0	0	82	17
6	MaxBlumenthal’	0	1	0	119	41
7	‘greenlinerzjm’	0	6	0	33	45
8	‘98Halima’	0	0	0	31	18
9	‘johnnygjokaj’	0	0	0	31	18
10	‘srhbutts’	39	3	0	0	16

In the table above, can observe that UM "freebsdgirl" falls into the "other_cyberbullying" class with 149 instances in comparison, it is also associated with the "gender" class in 7 instances and the "not_cyberbullying" class in 75 instances; this indicates that "freebsdgirl" is categorized as a bully in the "other_cyberbullying" type.

Similarly, UM "tayyoung_" has been associated with 957 instances in the "ethnicity" class, so the bully belongs to the class of cyberbullying type related to ethnicity, etc.

4.6.2 Users Mention Centrality Measures

Table 4.12 presents the top 10 user mentions in the dataset based on DC, BC, and CC. 'tayyoung_' holds the highest DC is 0.046, signifying a significant volume of mentions directed at this user. 'freebsdgirl' emerges as a critical player in mention connections with the highest BC and CC is 0.045 and 0.0831 simultaneously, suggesting her pivotal role in bridging different parts of the mention network. Moreover, 'freebsdgirl' efficiently reaches other users through mentions, as indicated by her leading Closeness Centrality score. These metrics provide valuable insights into the prominence and influence of specific users in the context of mentions within the dataset and indicate to the bully.

Table 4.12: *Top 10 Users Mention Centrality Measures*

seq.	Mention	DC	Mention	BC	Mention	CC
1	'tayyoung_'	0.046	freebsdgirl'	0.045	'freebsdgirl'	0.0831
2	'freebsdgirl'	0.035	'twitter'	0.025	'twitter'	0.0766
3	'MT8_9'	0.027	'MT8_9'	0.023	'Feminazi_Front'	0.0727

4	'MaxBlumenthal'	0.013	'realDonaldTrump'	0.02	'Brittany_Blade'	0.0725
5	'YesYoureSexist'	0.013	'PMOIndia'	0.015	'srhbutts'	0.0713
6	'srhbutts'	0.013	'nytimes'	0.012	'PolitiBunny'	0.0711
7	'realDonaldTrump'	0.012	'MaxBlumenthal'	0.011	'SwiftOnSecurity'	0.0711
8	'Spacekatgal'	0.011	'FoxNews'	0.011	'greyaesthetic'	0.0711
9	'wadhwa'	0.011	'YesYoureSexist'	0.01	'wadhwa'	0.0704
10	'a_man_in_black'	0.01	'Angry_Feminazi'	0.01	'PendragonTarot'	0.0703

In our study of the CBCD social network, several important metrics have examined to understand the roles of key individuals within the network. These metrics include DC, BC and CC. By looking at these metrics for each UM in the network, valuable insights can gain into how the network operates and make meaningful conclusions about the people involved.

Table 4.13 shows a particular group of users in the UM CBCD social network. Most of the members in this group are highly connected to others, as indicated by their high closeness centrality. Notably, "tayyoung_" stands out because it uniquely connects different people within the network, as seen through their significant BC.

Table 4.13: *Exclusive Circle for Users Mentions*

Node	**Degree Centrality**	**Betweenness Centrality**	**Closeness Centrality**
tayyoung_	0.0453	0.0024	0.0444
freebsdgirl	0.0349	0.044	0.0831
MT8_9	0.0268	0.0228	0.069
MaxBlumenthal	0.0129	0.0107	0.0471
YesYoureSexist	0.0126	0.0093	0.049
srhbutts	0.0123	0.0043	0.0713
realDonaldTrump	0.0111	0.0191	0.0639

Spacekatgal	0.0106	0.0037	0.0676
wadhwa	0.0103	0.0023	0.0704
a_man_in_black	0.0093	0.0016	0.0696
Feminazi_Front	0.0085	0.006	0.0727
SwiftOnSecurity	0.0077	0.001	0.0711
Brittany_Blade	0.0071	0.0045	0.0725
twitter	0.0048	0.0248	0.0766
FoxNews	0.0045	0.0101	0.0602
PMOIndia	0.0029	0.0148	0.0689
nytimes	0.0025	0.0119	0.0545
PolitiBunny	0.0023	0.0032	0.0711
PendragonTarot	0.0016	0.0006	0.0703
greyaesthetic	0.0012	0.0026	0.0711
Angry_Feminazi	0.0012	0.0092	0.0638

4.6.3 Hashtags

Table 4.14 displays the top 10 hashtags in the dataset, along with their corresponding frequencies. The hashtag 'mkr' is the most frequent, appearing 2524 times. Suggests that 'mkr' is a widely used and recurring hashtag within the dataset, likely associated with a specific topic, event, or trend. Other notable hashtags include 'notsexist,' 'islam,' 'mkr2015,' and 'blameonenotall.' The frequency of these hashtags reflects their prevalence in discussions or conversations within the dataset. Hashtags serve as a way to categorize and organize content on social media, making them essential for tracking and understanding trending topics and discussions.

Table 4.14: *Top 10 Hashtags Frequencies*

Seq.	Hashtag	Frequancy
1	'mkr'	2524
2	'notsexist'	168
3	'islam'	157
4	'mkr2015'	150
5	'blameonenotall'	127
6	'coon'	112
7	'isis'	100
8	'mileycyrus'	69
9	'stopwadhwa2015'	55
10	'128514'	54

Through this analysis evaluated hashtags by considering both their frequency (how frequently a hashtag is used) and stability (how consistently it maintains its frequency and thematic content over time). The hashtag categorized with the most frequent usage.

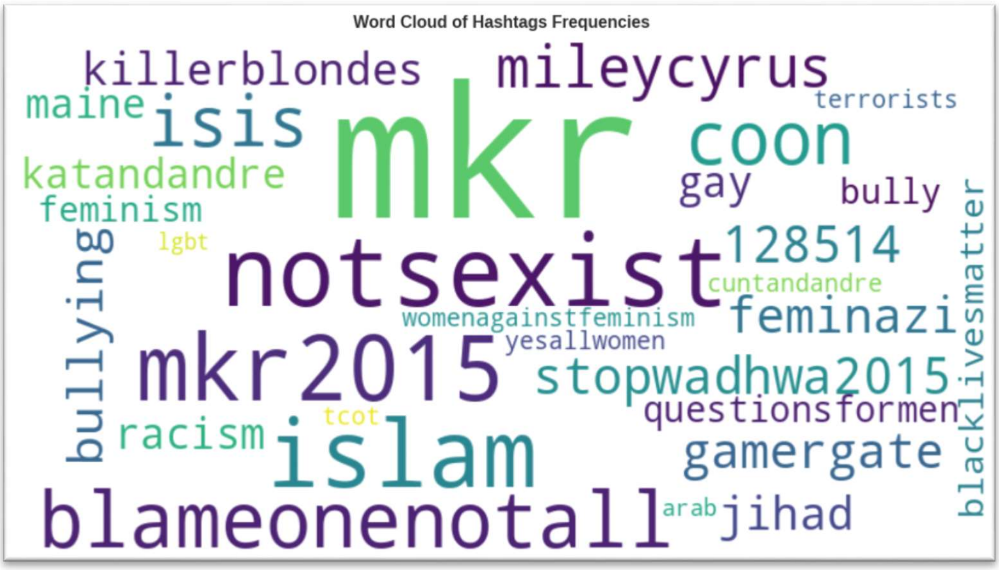


Figure 4.10: Word cloud Of Hashtags Frequency

As shown in Table 4.15, 'mkr' appears to be the most frequent hashtag in the dataset but the highest frequency in the 'not_cyberbullying.' Class is 1579, followed by "gender" and 'other_cyberbullying' so this indicates that it is primarily associated with non-bullying content. In contrast, 'notsexist' and "islam" are more consistently linked to specific bullying-related classes, such as 'gender' and 'religion,' respectively, making them potential candidates for hashtags associated with cyberbullying.

The connection lies in the understanding that high frequency, combined with consistency in thematic content over time, can signal the potential impact and significance of certain hashtags within specific contexts, which may help identify bullying-related content.

Table 4.15: *Top 10 Hashtags with Its Class Counts*

		Classes				
Seq.	**Hashtag**	**other_cyberbullying**	**gender**	**ethnicity**	**religion**	**not_cyber bullying**
1	'mkr'	278	630	0	2	1591
2	'notsexist'	0	163	0	0	5
3	'islam'	1	2	0	174	27
4	'mkr2015'	18	50	0	0	82
5	'blameonenotall'	109	17	0	0	0
6	'coon'	72	5	0	0	0
7	'isis'	4	2	0	60	38
8	'mileycyrus'	0	69	0	0	0
9	'stopwadhwa2015'	36	1	0	0	18
10	128514'	1	12	0	0	0

4.6.4 Hashtags Centrality Measures

Table 4.18 presents the top 10 hashtags in the dataset based on various centrality measures: DC, BC and CC. 'mkr' stands out with the highest DC, indicating a significant number of interactions involving this hashtag. 'mkr' also exhibits the highest BC, suggesting its pivotal role in connecting different parts of the hashtag network. Moreover, 'mkr' maintains efficient interactions with other hashtags, as indicated by its leading CC score. These metrics provide insights into the prominence and influence of specific hashtags within the dataset.

Table 4.18: *Top 10 Hashtags Centrality Measures*

Seq.	Hashtag	DC	Hashtag	BC	Hashtag	CC
1	mkr	0.2167	mkr	0.25	mkr	0.17
2	notsexist	0.0146	isis	0.07	24516	0.15
3	Islam	0.0134	4846	0.06	4846	0.15
4	mkr2015	0.013	24516	0.05	29205	0.14
5	blameonenotall	0.011	coon	0.04	1317	0.14
6	coon	0.0095	29205	0.03	15438	0.14
7	isis	0.0083	notsexist	0.03	isis	0.14
8	mileycyrus	0.006	bullying	0.03	15052	0.14
9	stopwadhwa2015	0.0048	blameonenotall	0.03	3422	0.14
10	gamergate	0.0045	gamergate	0.03	8326	0.14

In this thesis, a combination of centrality measures on hashtags have employed, as shown in Table 4.19, to help us understand how often they are used and how reliable they are for detecting cyberbullying (CBD).

When talking about "high centrality measures," this mean a hashtag is particularly important and used frequently. In this case, "notsexist" stands out as the second most significant hashtag after "mkr," with a frequency of 168, as indicated in Table 4.14.

Essentially, looking at how often hashtags are used and how consistently they point to cyberbullying content over time, and "notsexist" seems to be quite notable in this regard.

Table 4.17: Exclusive Circle for Hashtags

Node	DC	BC	CC
mkr	0.2168	0.2409	0.1644
notsexist	0.0146	0.028	0.0939
islam	0.0135	0.0194	0.1102
mkr2015	0.013	0.0007	0.1073
blameonenotall	0.0111	0.0227	0.1267
coon	0.0096	0.0342	0.1281
isis	0.0084	0.067	0.1334
mileycyrus	0.006	0.0071	0.0814
stopwadhwa2015	0.0048	0.0056	0.1117
gamergate	0.0046	0.0208	0.123
bullying	0.0042	0.0232	0.1201
24516	0.0013	0.0432	0.1432
4846	0.0006	0.0544	0.1422
29205	0.0005	0.0296	0.1377
8326	0.0005	0.0146	0.1311
3422	0.0005	0.0019	0.1313
1317	0.0005	0.0195	0.1348
15052	0.0003	0.004	0.133
15438	0.0002	0.0065	0.1339

CHAPTER FIVE

CONCLUSION AND FUTURE WORKS

5.1 Overview

This chapter presents two sections. In the first one, the conclusions obtained at in this thesis. After that, section 5.3 points out future work.

5.2 Conclusions

In the following, the conclusions of the proposed models are presented separately.

1. First model involved text mining, for the features extraction step, techniques TF-IDF with BoW and Word2Vec with PCA are employed, and for classification, four supervised ML models used. The RF with CBCD got results in terms of accuracy and precision, with minimal computational time, making it a prime candidate for thesis SNA model.

2. The second model detects cyberbullying by extracting new features related to user mentions and hashtags. Co-occurrence relationships and class counts were used to assess their significance in identifying cyberbullying patterns. These features were transformed into graphs, and centrality measures were calculated, including DC, BC, and CC. The combination of high centrality measures helps to detect influential users; the frequency of user mentions and the stability of hashtags over time allowed us to assess the effectiveness of our approach.

5.3 Future Work

1. A combination of sentiment analysis and natural language processing can help better understand the nature of cyberbullying content.

2. In SNA, using other Twitter terms like retweets, emotions, replies, and user IDs helps make an action for the bully user, such as blocking them from that platform.

3. Collaborating with social media platforms and organizations to implement real-time cyberbullying prevention measures would be helpful.

REFERENCES

[1] G. M. Abaido, “Cyberbullying on social media platforms among university students in the United Arab Emirates,” *Int. J. Adolesc. Youth*, vol. 25, no. 1, pp. 407–420, Dec. 2020, doi: 10.1080/02673843.2019.1669059.

[2] A. Ibtihaj and J. Alves-Foss, “Cyber_Bullying_and_Machine_Learning_A_Su.pdf.” International Journal of Computer Science and Information Security (IJCSIS). doi: https://doi.org/10.5281/zenodo.4249340.

[3] A. Wang and K. Potika, “Cyberbullying Classification based on Social Network Analysis,” in *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*, May 2021, pp. 87–95. doi: 10.31979/etd.9bn7-tq9h.

[4] Y. Peled, “Cyberbullying and its influence on academic, social, and emotional development of undergraduate students,” *Heliyon*, vol. 5, no. 3, p. e01393, Mar. 2019, doi: 10.1016/j.heliyon.2019.e01393.

[5] I. Alanazi and J. Alves-foss, “Cyber Bullying and Machine Learning : A Survey,” *International Journal if Computer Science and Information Security*, vol. 18, no. 10. pp. 1–8, 2020. doi: 10.5281/zenodo.4249340.

[6] B. Haidar, M. Chamoun, and F. Yamout, “Cyberbullying Detection: A Survey on Multilingual Techniques,” in *Proceedings - UKSim-AMSS 2016: 10th European Modelling Symposium on Computer Modelling and Simulation*, Nov. 2017, pp. 165–171. doi: 10.1109/EMS.2016.037.

[7] K. Das, S. Samanta, and M. Pal, “Study on centrality measures in social networks: a survey,” *Soc. Netw. Anal. Min.*, vol. 8, no. 1, 2018, doi: 10.1007/s13278-018-0493-2.

[8] V. Balakrishnan, S. Khan, and H. R. Arabnia, “Improving cyberbullying detection using Twitter users’ psychological features and machine learning,” *Comput. Secur.*, vol. 90, p. 101710, Mar. 2020, doi: 10.1016/j.cose.2019.101710.

[9] M. Nurek and R. Michalski, “Combining machine learning and social network analysis to reveal the organizational structures,” *Appl. Sci.*, vol. 10, no. 5, 2020, doi: 10.3390/app10051699.

[10] Y. J. Choi, B. J. Jeon, and H. W. Kim, “Identification of key cyberbullies: A text mining and social network analysis approach,” *Telemat. Informatics*, vol. 56, no. June, p. 101504, 2021, doi: 10.1016/j.tele.2020.101504.

[11] A. Wang, “Cyberbullying Classification based on Social Network Analysis,” San Jose State University, San Jose, CA, USA, 2021. doi: 10.31979/etd.9bn7-tq9h.

[12] M. I. Mahmud, M. Mamun, and A. Abdelgawad, “A Deep Analysis of Textual Features Based Cyberbullying Detection Using Machine Learning,” in *2022 IEEE Global Conference on Artificial Intelligence and Internet of Things, GCAIoT 2022*, 2022, no. December, pp. 166–170. doi: 10.1109/GCAIoT57150.2022.10019058.

[13] B. Ioannis and X. Christos, “Cyberbullying Detection through NLP & Machine Learning-MSc Dissertation,” University of Piraeus School, 2023. [Online]. Available: https://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/15298/Cyberbullying Detection through NLP %26 Machine Learning-MSc Dissertation.pdf?sequence=1&isAllowed=y

[14] N. Yuvaraj *et al.*, “Automatic detection of cyberbullying using multi-feature based artificial intelligence with deep decision tree classification,” *Comput. Electr. Eng.*, vol. 92, p. 107186, Jun. 2021, doi: 10.1016/j.compeleceng.2021.107186.

[15] P. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*, Second edi. New York, NY, 2019. [Online]. Available: https://lccn.loc.gov/2017048641

[16] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, “A Survey on Text Classification Algorithms: From Text to Predictions,” *Information*, vol. 13, no. 2, p. 83, Feb. 2022, doi: 10.3390/info13020083.

[17] T. Kanan, A. Aldaaja, and B. Hawashin, “Cyber-bullying and cyber-harassment detection using supervised machine learning techniques in Arabic social media contents,” *J. Internet Technol.*, vol. 21, no. 5, pp. 1409–1421, 2020, doi: 10.3966/160792642020092105016.

[18] R. Damarta, A. Hidayat, and A. S. Abdullah, “The application of k-nearest neighbors classifier for sentiment analysis of PT PLN (Persero) twitter account service quality,” *J. Phys. Conf. Ser.*, vol. 1722, no. 1, 2021, doi: 10.1088/1742-6596/1722/1/012002.

[19] A. Khatri and P. P, “Sarcasm Detection in Tweets with BERT and GloVe Embeddings,” in *Proceedings of the Second Workshop on Figurative Language Processing*, 2020, pp. 56–60. doi: 10.18653/v1/2020.figlang-1.7.

[20] P. M. Rahate and M. B. Chandak, “Text normalization and its role in speech synthesis,” *Int. J. Eng. Adv. Technol.*, vol. 8, no. 5 Special Issue 3, pp. 115–122, 2019, doi: 10.35940/ijeat.E1029.0785S319.

[21] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, “Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations,” *Organ. Res. Methods*, vol. 25, no. 1, pp. 114–146, Jan. 2022, doi: 10.1177/1094428120971683.

[22] R. Mohammed, J. Rawashdeh, and M. Abdullah, “Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results,” in *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, 2020, pp. 243–248. doi: 10.1109/ICICS49469.2020.239556.

[23] S. Mazumder, “5 Techniques to Handle Imbalanced Data For a Classification Problem,” *Analytics Vidhya*, 2021. https://www.analyticsvidhya.com/blog/2021/06/5-techniques-to-handle-imbalanced-data-for-a-classification-problem/#:~:text=Imbalanced data refers to those,very low number of observations.

[24] S. R. Spiegler, “M Achine L Earning for the a Nalysis of,” vol. 6, no. April, pp. 39–56, 2015.

[25] H. Abdulla and W. Awad, “Text Classification of English News Articles using Graph Mining Techniques.” pp. 926–937, 2022. doi: 10.5220/0010954600003116.

[26] E. Kazem and M. Hashim, “Arabic Sentiment Analysis for Determining Terrorism Supporters on Twitter Using Data Mining Techniques,” 2019.

[27] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi, “Sentiment analysis based on improved pre-trained word embeddings,” *Expert Syst. Appl.*, vol. 117, pp. 139–147, 2019, doi: 10.1016/j.eswa.2018.08.044.

[28] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations ofwords and phrases and their compositionality,” *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2013.

[29] U. Naseem, I. Razzak, S. K. Khan, and M. Prasad, “A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models,” *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 5, pp. 1–46, 2021, doi: 10.1145/3434237.

[30] A. El Mahdaouy, S. O. El Alaoui, and E. Gaussier, “Improving Arabic information retrieval using word embedding similarities,” *Int. J. Speech Technol.*, vol. 21, no. 1, pp. 121–136, 2018, doi: 10.1007/s10772-018-9492-y.

[31] T. Mikolov, G. Corrado, K. Chen, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” pp. 1–12.

[32] A. Pai, “An Essential Guide to Pretrained Word Embeddings for NLP Practitioners,” 2020.

[33] D. Jatnika, M. A. Bijaksana, and A. A. Suryani, “Word2Vec Model Analysis for Semantic Similarities in English Words,” in *Procedia Computer Science*, 2019, vol. 157, pp. 160–167. doi: 10.1016/j.procs.2019.08.153.

[34] J. Han, M. Kamber, and J. Pei, “Getting to Know Your Data,” in *Data Mining*, Elsevier, 2012, pp. 39–82. doi: 10.1016/B978-0-12-381479-1.00002-2.

[35] A. Wang and K. Potika, “Cyberbullying Classification based on Social Network Analysis,” in *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*, May 2021, pp. 87–95. doi: 10.31979/etd.9bn7-tq9h.

[36] D. L. Hansen, B. Shneiderman, and M. A. Smith, “Social Network Analysis,” in *Analyzing Social Media Networks with NodeXL*, Elsevier, 2011, pp. 31–50. doi: 10.1016/B978-0-12-382229-1.00003-5.

[37] R. Zafarani, M. A. Abbasi, and H. Liu, *Social media mining: An introduction*, vol. 9781107018853. 2014. doi: 10.1017/CBO9781139088510.

[38] M. Tsvetovat and A. Kouznetsov, *Social Network Analysis for Startups*. O’Reilly Media, Inc., 2011. [Online]. Available: https://www.oreilly.com/

[39] H. Tetsat, S. Puri, and B. Lookabaugh, “Developing a Machine Learning Model in Python,” in *Machine Learning and Data Science Blueprints for Finance*, O’Reilly Media, Inc., 2020.

[40] M. R. Belgaum *et al.*, “Enhancing the Efficiency of Diabetes Prediction through Training and Classification using PCA and LR Model,” *Ann. Emerg. Technol. Comput.*, vol. 7, no. 3, pp. 78–91, 2023, doi: 10.33166/AETiC.2023.03.004.

[41] A. M. Alduailaj and A. Belghith, “Detecting Arabic Cyberbullying Tweets Using Machine Learning,” *Mach. Learn. Knowl. Extr.*, vol. 5, no. 1, pp. 29–42, 2023, doi: 10.3390/make5010003.

[42] Soham Pathak, Antara Pawar, Shruti Taware, Sarthak Kulkarni, and Afsha Akkalkot, “A Survey on Machine Learning Algorithms for Risk-Controlled Algorithmic Trading,” *Int. J. Sci. Res. Sci. Technol.*, pp. 1069–1089, Jun. 2023, doi: 10.32628/IJSRST523103163.

[43] S. Keleş, A. Günlü, and İ. Ercanli, “Estimating aboveground stand carbon by combining Sentinel-1 and Sentinel-2 satellite data: a case study from Turkey,” in *Forest Resources*

Resilience and Conflicts, Elsevier, 2021, pp. 117–126. doi: 10.1016/B978-0-12-822931-6.00008-3.

[44] A. Muneer and S. M. Fati, “A comparative analysis of machine learning techniques for cyberbullying detection on twitter,” *Futur. Internet*, vol. 12, no. 11, pp. 1–21, 2020, doi: 10.3390/fi12110187.

[45] L. Sunitha and M. B. Raju, “Multi-class classification for large datasets with optimized SVM by non-linear kernel function,” *J. Phys. Conf. Ser.*, vol. 2089, no. 1, 2021, doi: 10.1088/1742-6596/2089/1/012015.

[46] O. A. Montesinos López, A. Montesinos López, and J. Crossa, *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. 2022. doi: 10.1007/978-3-030-89010-0.

[47] M. F. Hassan and M. E. Manaa, “Big Data Processing with Hadoop and Data Mining,” in *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Jun. 2022, pp. 1–8. doi: 10.1109/HORA55278.2022.9800085.

[48] Y. Liu, Y. Wang, and J. Zhang, “New Machine Learning Algorithm: Random Forest,” 2012, pp. 246–252. doi: 10.1007/978-3-642-34062-8_32.

[49] M. H. U. Rahman, “Cyberbullying Detection using Natural Language Processing,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 5, pp. 5241–5248, 2022, doi: 10.22214/ijraset.2022.43683.

[50] S. Hota and S. Pathak, “KNN classifier based approach for multi-class sentiment analysis of twitter data,” *Int. J. Eng. Technol.*, vol. 7, no. 3, pp. 1372–1375, 2018, doi: 10.14419/ijet.v7i3.12656.

[51] J. Han, M. Kamber, and J. Pei, “Chapter 9: Graph Mining, Social Network Analysis, and Multirelational Data Mining,” in *Data mining: concepts and techniques*, 2006, pp. 535–589. [Online]. Available: http://books.google.com/books?hl=en&lr=&id=AfL0t-YzOrEC&oi=fnd&pg=PP2&dq=Data+Mining:+Concepts+and+Techniques&ots=Uv-WrRflz9&sig=E_20H417umQmqnf_jM9m9DyNRf0

[52] A. Rajaraman and J. D. Ullman, *Mining of massive datasets*, 3rd editio., vol. 9781107015. Stanford, 2011. doi: 10.1017/CBO9781139058452.

[53] D. Antonakaki, P. Fragopoulou, and S. Ioannidis, “A survey of Twitter research: Data model, graph structure, sentiment analysis and attacks,” *Expert Syst. Appl.*, vol. 164, p. 114006, Feb. 2021, doi: 10.1016/j.eswa.2020.114006.

[54] Q. X. Jinhuan Wang, Pengtao Chen, Bin Ma, Jiajun Zhou, Zhongyuan Ruan, Guanrong Chen, “Sampling Subgraph Network with Application to Graph Classification.” 2021. doi: https://doi.org/10.48550/arXiv.2102.05272.

الخلاصة

أدى النمو السريع لوسائل التواصل الاجتماعي إلى ظهور أشكال جديدة من التنمر الالكتروني. أصبحت منصات التواصل الاجتماعي مثل فيسبوك وتويتر ويوتيوب مصدر قلق كبير للأفراد والمنظمات والمجتمع ككل. يعد الكشف المبكر عن التسلط عبر الإنترنت واعتراضه أمرا بالغ الأهمية للتخفيف من آثاره الضارة.

تضمن النظام المقترح نموذجين. تضمن النموذج الأول مجموعتين من البيانات متعددة التصنيفات وعمل مع التنقيب عن النصوص لتصنيف التغريدات إلى تصنيفات متعددة باستخدام تقنيات مختلفة. استخدم النموذج الثاني تحليل الشبكة الاجتماعية social network analysis (SNA) للكشف عن المستخدمين المؤثرين الذين نشروا التنمر في المجتمعات ومحتوى التنمر المرتبط به.

في النموذج الأول ، العديد من التقنيات المستخدمة في خطوة استخراج الميزات هي TF-IDF مع Bow و Word2Vec للتصنيف ، يتم استخدام أربعة خوارزميات التعلم الآلي ، Random Forest (RF), Support Vector Machine (SVM), K-nearest neighbors (KNN), and Naïve Bayes (NB).واستخدم النموذج الثاني ثلاثة مقاييس مركزية centrality measures: degree centrality (DC), betweenness centrality (BC), and closeness centrality (CC).

أظهرت نتائج النموذج الأول فعالية مجموعة البيانات الأولى، "Cyberbullying Classification Dataset"، دقة ومعدلات الدقة 93 ٪ و 87 ٪ على التوالي. بينما حصلت مجموعة البيانات الثانية، "Cyber bullying Types Dataset"، على نتائج بدقة ومعدلات دقة بلغت 89 ٪ و 90 ٪ على التوالي ، أدت هذه النتائج إلى اختيار " Cyberbullying Classification Dataset "كبيانات مناسبة لتحليل الشبكات الاجتماعية social network analysis (SNA) .

استنتجت الدراسة بأن كشف social network analysis (SNA) عن رؤى قيمة حول اكتشاف التسلط عبر الإنترنت ، مع التركيز بشكل خاص على الإشارات المتكررة للمستخدمين (user mentions) (المستخدمين المؤثرين) والمقاييس المركزية العالية (centrality measures) كمؤشرات موثوقة. وأن استقرار علامات التصنيف (hashtags) بمرور الوقت أيضا دورا مهما في تحديد المحتوى المرتبط بالتنمر.

جامعة كربلاء
كلية علوم الحاسوب وتكنولوجيا المعلومات
قسم علوم الحاسوب

تصنيف التنمر الالكتروني واكتشافه في تويتر بإستخدام تقنيات تنقيب البيانات

رسالة ماجستير

مقدمة الى مجلس كلية علوم الحاسوب وتكنولوجيا المعلومات / جامعة كربلاء
وهي جزء من متطلبات نيل درجة الماجستير في علوم الحاسوب

كتبت بواسطة

فاطمة نادي علي حسين

بإشـراف

أ.م.د هبة جبار العقابي

1445 هـ **2024 م**